

Vom Fachbereich Maschinenwesen der Universität Duisburg-Essen
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs genehmigte Dissertation

Konzeption und Realisierung einer Supply Chain Management-orientierten
Anwendungsintegration im mittelständischen Automotive-Umfeld

Vorgelegt von

Dipl.-Ing. Christoph Kornek, Fulda

Referent: Priv.-Doz. Dr.-Ing. F. Lobeck
Korreferent: Univ. Prof. Dr.-Ing. D. Bergers

Tag der mündlichen Prüfung: 14.10.2004

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als CIO der ttb-Gruppe mit Sitz in Eichenzell. Den Vorständen Herrn H. Thiele und besonders Herrn W. Braun danke ich für die Förderung dieser Arbeit sowie für den dabei gewährten Freiraum.

Herrn Univ.-Prof. Dr.-Ing. H. J. Stracke danke ich für die Unterstützung während meiner promotionsvorbereitenden Studien und für die Anregung und Förderung dieser Arbeit.

Herrn Priv.-Doz. Dr.-Ing. F. Lobeck danke ich für das meiner Arbeit entgegengebrachte Interesse, die damit verbundenen positiven Anmerkungen und für die Übernahme des Erstgutachtens.

Herrn Univ.-Prof. Dr.-Ing. D. Bergers danke ich für die fachliche Unterstützung und die Übernahme des Korreferats.

Herrn Univ.-Prof. Dr.-Ing. E. von Lavante danke ich für die Übernahme des Prüfungsvorsitzes.

Danken möchte ich auch meinem Kollegen Herrn M. Spork von der Thermoplast Technik GmbH & Co. KG für seine Unterstützung und seine Bereitschaft, meine Konzepte mit ihm diskutieren zu können.

Meinen Eltern danke ich dafür, dass Sie mir meinen beruflichen Werdegang ermöglichten und mir auch in schwierigen Situationen stets zur Seite standen.

Fulda, im Oktober 2004

Christoph Kornek

Inhaltsverzeichnis

1	Einleitung.....	1
2	Analyse des Ist-Zustandes	6
2.1	Unternehmensinterne Situation	7
2.2	Globalisierung und Preisdruck.....	9
2.3	Produkt- und Prozessinnovation.....	14
2.4	Konzentration und Konsolidierung.....	17
2.5	Customer Relation	20
2.6	Transport – und Lagerlogistik	23
2.7	Informations- und Kommunikationsbarrieren.....	27
2.8	Fazit.....	34
3	Anforderungen für eine Anwendungsintegration	38
3.1	Anforderungen aus der Sicht der Unternehmen.....	38
3.2	Anforderungen an die Kommunikation	39
3.3	Anforderungen an die Schnittstellen.....	41
3.4	Anforderungen im Zusammenhang mit der Integration	42
3.5	Zusammenfassung der Anforderungen	47
4	Grundlagen der eingesetzten Informationstechnik.....	49
4.1	Verfügbare EAI-Systeme	49
4.2	Klassifizierung von EAI-Systemen.....	51
4.3	Programmierungsumgebung für das Konzept.....	53
5	Konzept für eine Anwendungsintegration.....	56
5.1	Allgemeines	56
5.2	Analyse der vorhandenen EAI-Systeme.....	56
5.2.1	TIBCO: Active Enterprise.....	57
5.2.2	Extricity. Inc.: AllianceManager.....	59
5.2.3	Microsoft: BizTalk Server 2002.....	62
5.2.4	Vitria: BusinessWare	64
5.2.5	Data Mirror: Constellar Hub	67
5.2.6	CrossWorlds: CrossWorlds.....	69
5.2.7	SeeBeyond: e*Gate / e*Xchange	73
5.2.8	New Era of Networks (NEON): e-Biz Integrator.....	75
5.2.9	BEA Systems: eLink / Tuxedo / WebLogic	77
5.2.10	iPlanet: Forté Fusion.....	79
5.2.11	NEON Systems: iWave.....	82
5.2.12	Mercator Software: Mercator	84

5.2.13	SunGard Business Integration: MINT	86
5.2.14	IBM: MQSeries / MQIntegrator / MQWorkflow	88
5.2.15	!Candle: ROMA.....	89
5.2.16	Compuware: UNIFACE.....	91
5.2.17	WebMethods: WebMethods	93
5.3	Beurteilung der EAI-Systeme zwecks Eignung für das Konzept	95
5.4	LAMP als Grundlage für die Anwendungsintegration	98
5.5	Detaillkonzepte	111
5.5.1	VMI	111
5.5.2	ATP.....	117
5.5.3	Portallösung.....	120
5.5.4	Prozessdatenaustausch/EDI	122
6	Realisierung	128
6.1	Wahl des Unternehmens	128
6.2	Software- und Systemumgebung	129
6.3	Programmtechnische Umsetzung.....	131
6.3.1	VMI	131
6.3.2	ATP.....	137
6.3.3	Portallösung.....	139
6.3.4	Prozessdatenaustausch / EDI	141
6.4	Diskussion der erzielten Ergebnisse	143
7	Zusammenfassung und Ausblick.....	148
8	Abbildungsverzeichnis.....	151
9	Tabellenverzeichnis.....	154
10	Literaturverzeichnis	155
11	Abkürzungsverzeichnis	166
12	Anhang	171
12.1	Tabellenbeschreibungen der Tabellen auth und rights der Datenbank konsi	171
12.2	PHP-Programmskript zum VMI.....	171
12.3	PHP-Programmskript zum Auslesen der Oracle-Datenbank.....	180
12.4	PHP-Programmskript für die Einbindung von Bildinformationen	182
12.5	PHP-Programmskript für die Aufbereitung der Lagerbestandsinformationen aus der ERP-Datenbank als XML-Nachricht	183
12.6	PHP-Programmskript für den Versand der XML-Nachricht als Email.....	184

1 Einleitung

In der Automobilindustrie sind seit einigen Jahren Entwicklungen zu beobachten, die die gesamte Branche zu Veränderungen zwingen: Globalisierung, steigender Preisdruck sowie anspruchsvolles Kundenverhalten sind nur einige Schlagworte, hinter denen sich die Notwendigkeit zur Umsetzung neuer Prozessmodelle entlang der gesamten automobilen Wertschöpfungskette verbirgt. Diese neuen Prozessmodelle orientieren sich unmittelbar an der Wertschöpfung selbst, sie beschreiben die Vorgänge des Liefers, Fertigungs und Versendens der Ware von einem Lieferanten zum nächsten und werden unter dem Begriff **SCM**¹ zusammengefasst. Anders als bisher werden dadurch nicht nur unternehmensinterne Geschäftsprozesse beschrieben, sondern auch -externe mit dem Ziel, die Lieferkette vom ersten Lieferanten bis zum Automobilhersteller und schließlich bis hin zum Endkunden transparent zu gestalten und zu steuern. Ein zentrales Problem stellt hierbei die Kommunikation zwischen den größtenteils isolierten Teilprozessen entlang der Supply Chain dar, da die beteiligten Prozesssysteme häufig nicht kompatibel untereinander sind und nur mit schwankenden Zeitverzögerungen die relevanten Prozessinformationen weitergeben können.

Grundlage für ein effizientes Supply Chain Management ist daher die Verknüpfung dieser Systeme, um Informationen verzögerungsfrei in Echtzeit zu verarbeiten und weiterzuleiten, was insbesondere für die empfindlichen Lieferketten der Automobilindustrie von besonderer Bedeutung ist. Die hierfür bereitzustellenden Funktionalitäten sind wesentliche Merkmale der **EAI**², welche die Verknüpfungen zwischen den in den Unternehmen vorhandenen Softwarelösungen herstellt und den direkten Datenaustausch zwischen den einzelnen Softwareanwendungen ermöglicht. Ausgehend von der Integration der Systeme in der internen Unternehmensstruktur können damit auch die Systeme jenseits der eigenen Systemgrenzen integriert werden. Dadurch wird effizientes Supply Chain Management mit externen Partnern erst möglich.

Ausgangspunkt für die Formulierung eines Konzeptes ist ein allgemeines Anwendungs- bzw. Integrationsszenario der Automobilindustrie, wie es in der Abbildung 1.1 dargestellt ist. Während eine Vielzahl von isolierten unternehmensinternen Anwendungen existieren und

¹ Supply Chain Management

² Enterprise Application Integration

ein eigenes inneres System bilden (**CAD**³, **ERP**⁴, **BDE**⁵ usw.), findet an den Grenzen dieses Systems die Kommunikation zur Außenwelt und damit zu äußeren Systemen statt (Automobilhersteller, Zulieferer und weitere Partner). Auch die Kommunikation zu verteilten Standorten eines Unternehmens findet hier statt.

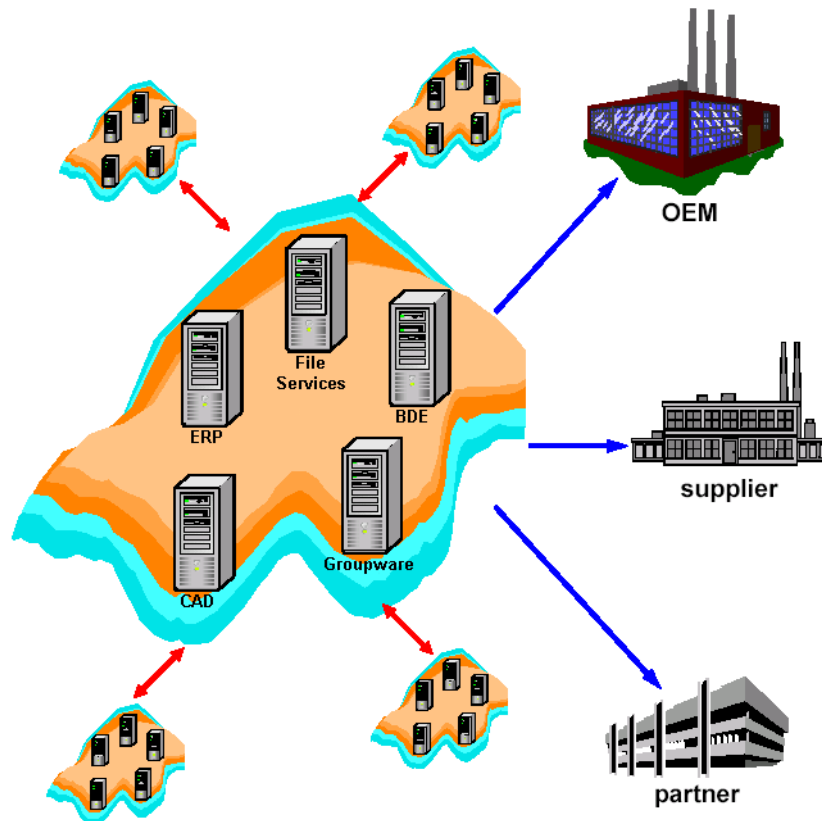


Abbildung 1.1: IT-Szenario in der Automobilindustrie ohne EAI

Damit nun die Kommunikation der inneren mit den äußeren Systemen möglich wird, ist ein Integrationssystem erforderlich, das mit allen beteiligten Systemen kommunizieren kann. Zentraler Bestandteil einer Integrationslösung ist daher eine EAI-Plattform, über die **IT**⁶-Anwendungen und Geschäftsprozesse intern und extern miteinander verbunden werden (vgl. Abbildung 1.2). Die bisherige Technologie, welche die Grundlage für diese Verbindungen bildet, wird auch Middleware genannt [1]. Während Middleware ursprünglich den

³ Computer Aided Design

⁴ Enterprise Ressource Planning

⁵ Betriebsdatenerfassung

⁶ Informationstechnologie

Datenaustausch zwischen den verschiedensten **API**⁷s einzelner Anwendungen bewerkstelligte, wird sie in einem EAI-System dazu verwendet, Informationen zwischen den Anwendungen und dem EAI-System zu transformieren und zu transportieren.

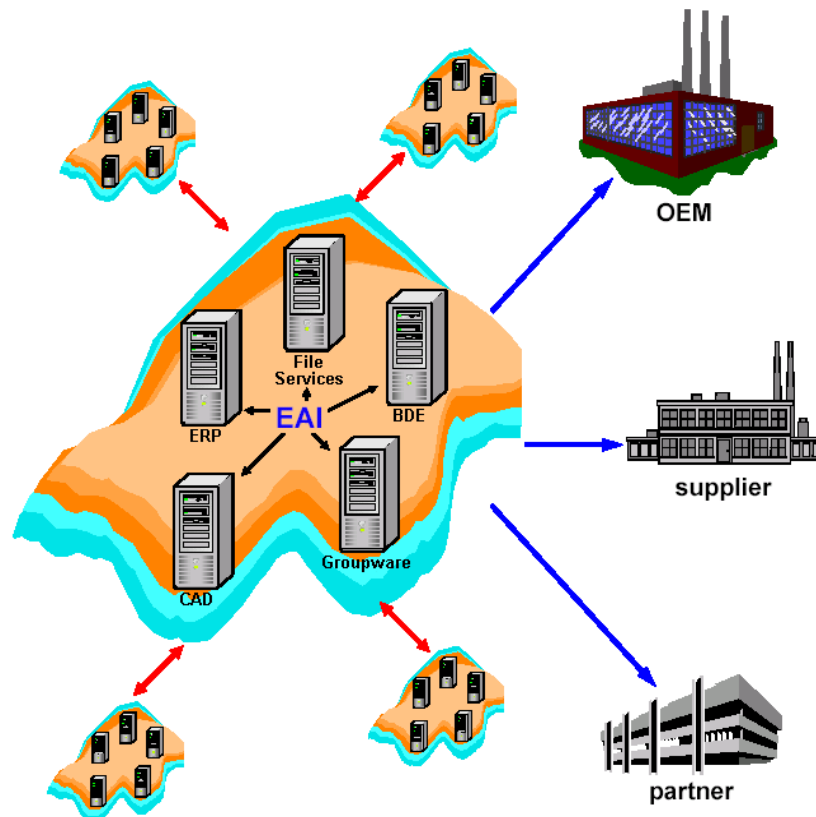


Abbildung 1.2: EAI-Plattform zur Integration der IT-Anwendungen

Insbesondere bei der Transformation wird auf ein Repository⁸ zurückgegriffen, eine Art Schablone, in der exakt definiert ist, aus welchen Daten beispielsweise ein Objekt „Kunde“ aufgebaut ist. Das Repository stellt Metadaten zur Verfügung und wird für jedes EAI-Umfeld individuell definiert. Es stellt eine Besonderheit der EAI-Systeme dar, die in der früheren Middleware nicht enthalten war.

Die spezifischen Integrationsanforderungen bestimmen, ob ein Repository benötigt wird oder nicht. Einfache EAI-Ansätze kommen auch ohne ein zentrales Repository aus, wenngleich die Notwendigkeit mit dem Grad der Integration schnell ansteigt. In früheren

⁷ Application Programming Interface

⁸ zentrale Datenbank (Metadaten-Repository), in der die Informationen und Regeln über die Datenquellen und -ziele, die Transformationsregeln und die Datenflussregeln gespeichert sind

Integrationsszenarien wurden so genannte **ETL**⁹-Tools eingesetzt, die zur regelmäßigen Aktualisierung der Daten eines Data Warehouses oder einer **OLAP**¹⁰-Applikation verwendet wurden. Die Daten müssen zunächst aus den Quellsystemen extrahiert, dann auf Konsistenz überprüft, gegebenenfalls angepasst und schließlich in die Datenbank des Data Warehouses geladen werden.

Es existiert eine Reihe von EAI-Systemen am Markt, die sich einerseits durch umfangreiche Funktionalitäten auszeichnen, andererseits jedoch hohe Lizenz- und Implementierungskosten aufweisen, die u. a. für die mittelständische Zulieferindustrie hohe Hürden für die Umsetzung von EAI- und den daraus abgeleiteten SCM-Konzepten darstellen. Ziel dieser Arbeit ist es, eine EAI-Anwendung zu beschreiben, mit der es schnell und kostengünstig möglich ist, SCM-Prozesse in der mittelständischen Automobilindustrie umzusetzen.

In der hier vorliegenden Arbeit wird zunächst der Ist-Zustand bezüglich der Informationsverarbeitung am Beispiel der Automobilzulieferindustrie einer kritischen Betrachtung unterzogen und insbesondere die mit der Globalisierung verbundenen Änderungen auf die Wertschöpfungskette dargestellt, und zwar auch unter der Berücksichtigung der Auswirkungen auf die Produkt- und Prozessinnovation im Zusammenhang mit der Hersteller-Lieferanten-Beziehung. Aus den so ermittelten Defiziten wird dann ein Forderungskatalog formuliert, der die notwendigen Verbesserungen bezüglich der Kommunikation, der Integration und der dafür notwendigen Schnittstellen beinhaltet. Für die Erfüllung dieser Forderungen werden anschließend die momentan verfügbaren Komponenten der Informationstechnologie kurz beschrieben.

Die Formulierung des Konzeptes für die Anwendungsintegration beginnt mit einer umfangreichen Analyse der auf dem Markt verfügbaren EAI-Systeme. Die Bewertung zeigt eindeutig, dass diese EAI-Systeme aus verschiedenen Gründen für den mittelständischen Markt noch nicht geeignet sind. Dies führt zu einem weiteren Ansatz für eine EAI-Anwendung, die auf Standardkomponenten beruht, die in den meisten mittelständischen Unternehmen bereits vorhanden sind. Für diesen Ansatz werden dann die Detailkonzepte einschließlich der notwendigen Anpassungsprogrammierung detailliert vorgestellt.

Die Machbarkeit des Konzeptes wird am Beispiel von ausgewählten Geschäftsprozessen eines mittelständischen Unternehmens aus der Automobilzulieferindustrie nachgewiesen. In

⁹ Extraktion, Transformation und Laden

¹⁰ Online Analytical Processing

einer Kosten-Nutzen-Analyse werden die erzielten Ergebnisse abschließend diskutiert. Danach erfolgt eine kurze Betrachtung der zu vertiefenden Aspekte der hier entwickelten Anwendungsintegration sowie ein Ausblick auf künftige Entwicklungsrichtungen.

Die in der Arbeit verwendeten Abkürzungen sind in einem Abkürzungsverzeichnis wiedergegeben. Sie sind bei der jeweils ersten Verwendung innerhalb des Textes fettgedruckt dargestellt sowie in Fußnoten erklärt.

2 Analyse des Ist-Zustandes

Die Automobilindustrie befindet sich im Wandel, Automobilhersteller (**OEM¹¹s**) und die überwiegend mittelständischen Zulieferer stehen vor großen Herausforderungen. Ausgehend von der Markt- und Wettbewerbsentwicklung, die durch Faktoren wie Preissteigerungen bei Rohstoffen, Überkapazitäten und globalen Veränderungen geprägt sind, entwickeln die Automobilunternehmen vielfältige Strategien von der Globalisierung über Kooperationen bis hin zum Gedanken eines „Weltautos“, um den künftigen Erwartungen gerecht zu werden [2]. An die Zulieferer stellen die Marktveränderungen neue Anforderungen, die sich in einem stetigen Preisdruck und der Notwendigkeit zur globalen Präsenz widerspiegeln (vgl. Abbildung 2.1).

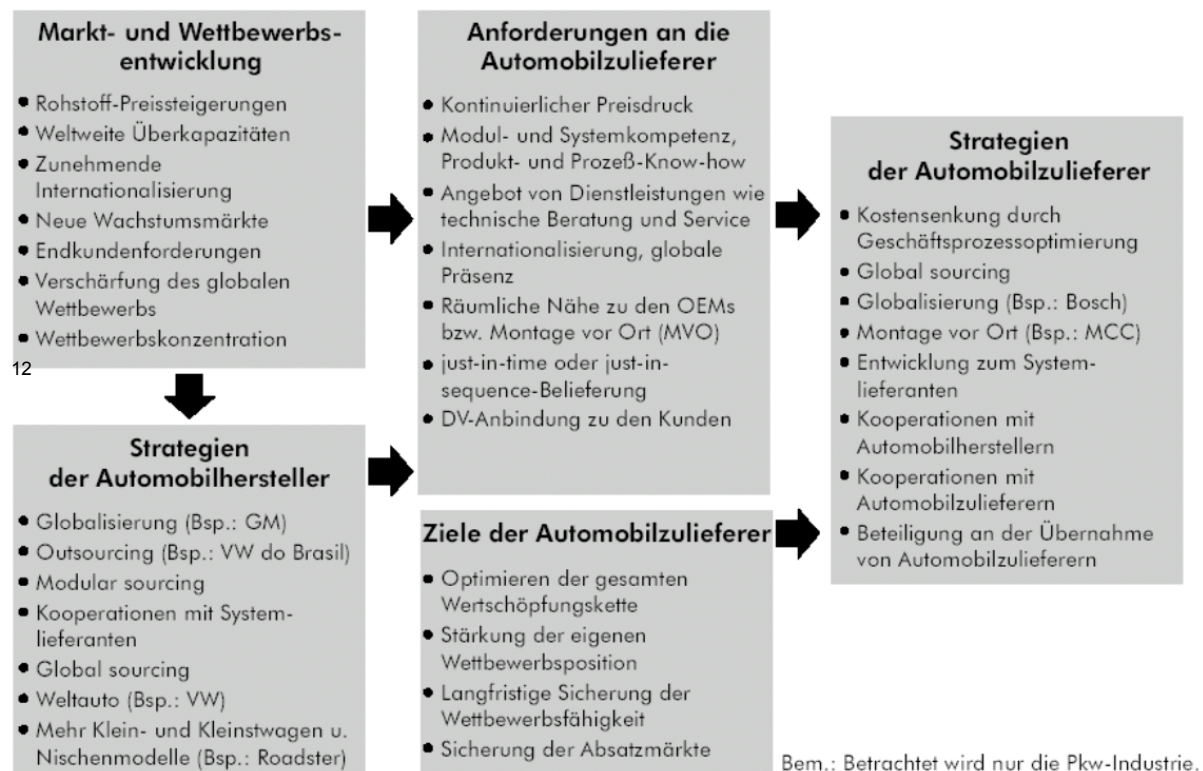


Abbildung 2.1: Die Automobilindustrie im Wandel [2]

¹¹ Original Equipment Manufacturer

¹² Micro Compact Car Smart GmbH, Hersteller des Pkw-Modells "Smart"

Im Rahmen der Erweiterung der Produkt- und Prozesskompetenz gewinnen dabei auch Logistikkonzepte wie **JIT**¹³ oder **JIS**¹⁴ zunehmende Bedeutung, diese werden bereits heute in großem Umfang praktiziert. Hieraus abgeleitet werden die Zulieferer künftig gezwungen sein, ihren gesamten Wertschöpfungsprozess zu optimieren, um die erforderlichen Zeit- und Terminvorgaben durch punktgenaues Fertigen und Versenden der von ihnen gelieferten Ware erfüllen zu können.

Auch grundsätzliche Unternehmensziele wie die Verbesserung der Wettbewerbsfähigkeit oder die Sicherung der Absatzmärkte erlangen wieder verstärkt Bedeutung. Damit werden die strategisch erforderlichen Aktionen dadurch geprägt, Kostensenkungspotentiale zu erkennen und zu nutzen, die Globalisierung auch auf Zulieferebene kontinuierlich voranzutreiben, mehr Kundennähe aufzubauen sowie Kooperationen und Beteiligungen mit Herstellern und anderen Zulieferern anzustreben.

Die folgende Betrachtung des Ist-Zustandes soll zunächst die beschriebenen Probleme der Automobilindustrie näher thematisieren und dabei die Schwerpunkte der mittelständischen Unternehmen herausarbeiten, um damit in den weiteren Kapiteln dieser Arbeit Ansatzpunkte zur Problemlösung beschreiben zu können.

2.1 Unternehmensinterne Situation

Mittelständische Unternehmen besitzen hinsichtlich ihrer Organisation und ihrer strategischen Ausrichtung Charakteristiken, die häufig unternehmensspezifisch sind und als „historisch gewachsen“ beschrieben werden [3]. So ergibt sich üblicherweise ein heterogenes, strukturelles Gesamtbild mittelständischer Unternehmen, das beispielsweise durch Wachstum, Akquisition oder spezifische Kundenanforderungen entstanden ist. Die Einbindung werksübergreifender, logistischer Konzepte spielt dabei eine untergeordnete Rolle, der Fokus ist hauptsächlich auf das Kerngeschäft gerichtet, das i. d. R. die Fertigung von Bauteilen oder Bauteilsystemen mit vergleichsweise hoher Fertigungstiefe darstellt. Dem Verständnis nach konzentriert sich hier die Wertschöpfung, so dass, katalysiert durch eine werkszentrische Sichtweise, typischerweise keine Betrachtung der gesamten Supply Chain vor und nach den eigenen Lieferanten und Kunden stattfindet.

¹³ Just In Time

¹⁴ Just in Sequence

Die Ursache für die daraus resultierenden Probleme in einer Kooperationsbeziehung zwischen zwei Unternehmen ist überwiegend in der Organisation der Zusammenarbeit begründet, denn die Prozesskette zur Entwicklung eines Produkts wurde bisher nur innerhalb des eigenen Unternehmens optimiert [4]. Unternehmensübergreifende Konzepte zur Synchronisation der Abläufe zwischen Hersteller und Zulieferer fehlen jedoch. Einerseits ist die Vertrauensbasis noch nicht ausreichend groß, um detaillierte Informationen über die eigenen Organisationsstrukturen an Geschäftspartner und Kunden weiterzugeben, andererseits verfügen die Unternehmen oftmals nicht über die technischen Möglichkeiten, diese Informationen zur Verfügung zu stellen. Durch die teilweise sehr weitgehende prozessuale Verzahnung von Geschäftsinteressen zwischen eigenem Unternehmen, Kunden und Lieferanten dürfte hier jedoch ein wesentlicher Optimierungsansatz liegen.

Führende Unternehmen haben daher damit begonnen, ihre funktionale Unternehmensorganisation in eine prozessorientierte Form umzugestalten. Eine prozessorientierte Organisation führt im Vergleich zu traditionellen Organisationsformen zu weniger Hierarchiestufen und zu mehr Handlungsmöglichkeiten für die Mitarbeiter bzw. Organisationseinheiten [5]. Auch werden Schnittstellenprobleme, wie sie häufig in funktional gegliederten Organisationen zu beobachten sind, in einer prozessorientierten Organisation durch die Orientierung am Gesamtergebnis des Prozesses vermieden [6]. Neue Ansätze zur Organisation von Unternehmen empfehlen in diesem Zusammenhang die Ausrichtung des Unternehmens an die zur Erstellung der Leistung nötigen Prozesse [7]. Damit wechselt die Perspektive, unter der das Unternehmen betrachtet wird, von der vertikalen Gliederung in Hierarchieebenen zur horizontalen Gliederung in Wertschöpfungsketten bzw. von der Funktionsoptimierung zur Prozessoptimierung [4]. Die Abbildung 2.2 illustriert diese Situation [8].

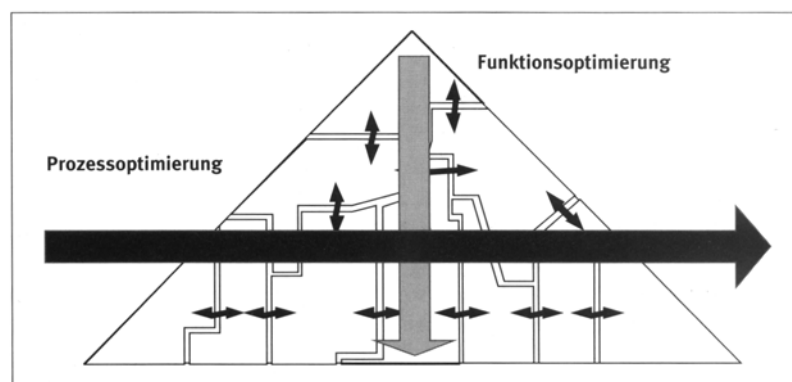


Abbildung 2.2: Prozessoptimierung versus Funktionsoptimierung [8]

Neben Schwächen im Aufbau der Unternehmensorganisation ist vielfach eine dem Marktumfeld nicht angepasste Strategie für das Scheitern von Unternehmen verantwortlich. Kern einer erfolgreichen Unternehmensstrategie stellt die Dominanz des Unternehmens in einem wohl definierten, fokussierten Geschäftsfeld dar, wodurch die Begriffe Dominanz und Fokus als Schlüsselfaktoren für dauerhaft profitables (nachhaltiges) Wachstum besondere Bedeutung besitzen [9]. Aktuelle Untersuchungen zufolge betreiben 78 Prozent der nachhaltig Wert schaffenden Unternehmen ein einziges, zentrales Kerngeschäft, 17 Prozent betreiben mehrere Kerngeschäfte und 5 Prozent sind ein Konglomerat mit mehreren Geschäften und Segmenten [9]. Ein richtig definiertes Kerngeschäft stellt insofern die Voraussetzung für eine erfolgreiche Strategie dar, die infolge von dynamischen Marktentwicklungen wie Technologiewechsels oder staatlichen Regulierungen eine regelmäßige Anpassung in der Definition des Kerngeschäfts erfahren muss, häufig jedoch nicht erfährt.

2.2 Globalisierung und Preisdruck

Während die unternehmensinterne Gestaltung von Prozessen und Strategiedefinitionen noch im Einflussbereich der Unternehmen liegen, gibt es darüberhinaus unternehmerische Rahmenbedingungen, die im Wesentlichen vom Gesamtmarkt bestimmt werden. So prägten ausgehend von einer Studie des **MIT**¹⁵ Begriffe wie Lean Production¹⁶ und Toyotismus¹⁷ bereits Anfang der neunziger Jahre die Automobilindustrie. Das auf dieser Studie aufbauende Werk „Die zweite Revolution in der Autoindustrie“ (engl.: The Machine That Changed the World) [10] gilt als „Bibel der Schlanken Produktion“ und beschreibt die Überlegenheit japanischer Automobilbauer gegenüber ihren weltweiten Konkurrenten, die sich bislang an den Konzepten der wissenschaftlichen Betriebsführung nach Taylor orientiert hatten. Hierbei erfolgte eine Zergliederung der Arbeit in sich ständig wiederholende, standardisierte Teilarbeitsgänge; dispositive und ausführende Arbeiten wurden organisatorisch getrennt, der Arbeiter erhielt Produktionsanweisungen und wurde fremdkontrolliert [11].

¹⁵ Massachusetts Institute of Technology

¹⁶ Schlanke Produktion, auch: Schlankes Unternehmen, zur gleichzeitigen Verbindung hoher Produktivität mit hoher Qualität [15]

¹⁷ Managementstil japanischer Automobilhersteller: flexible Bildung von Arbeitsteams, die ein bestimmtes Problem lösen

Im Wesentlichen wurden mit Lean Production Wege aufgezeigt, wie die Automobilproduktion mit weniger Personal noch schneller, profitabler und zielgerichteter durchgeführt werden sollte, wozu Maßnahmen wie Verringerung der Belegschaften zu einer „schlanken“ Struktur, geringere Lohnerhöhungen, neue Arbeitsorganisationen und verringerte Fertigungstiefen beitragen sollten [12]. Anschaulich ließ sich dies am Beispiel des japanischen Automobilherstellers Toyota verfolgen, dem drittgrößten Automobilhersteller der Welt, der im Jahr 1991 das größte und vermögendste japanische Automobilunternehmen und zugleich das gewinnbringendste japanische Unternehmen überhaupt darstellte [13].

Diese Entwicklung war Teil der Globalisierung der gesamten Automobilindustrie, die massive Auswirkungen auf Produktion, Modelle, Preise und Distribution vieler Hersteller hatte und bis heute hat [14]. Der Begriff der Globalisierung ist dabei in den letzten Jahren zu einem Modebegriff geworden, der meist als Schlagwort in der Diskussion um Standorte verwendet wird, als Definition findet sich auch „die zunehmende weltweite Vernetzung ökonomischer Aktivitäten“ [16]. Aktuelle Beispiele hierfür sind die Osterweiterung der Europäischen Union oder der **NATO**¹⁸. Wesentliche Kennzeichen der Globalisierung sind

- die zunehmende Mobilität der Produktionsfaktoren,
- das standortunabhängige Angebot von Dienstleistungen,
- der Anstieg der internationalen Finanzströme,
- die Zunahme von lukrativen und „sicheren“ Standorten sowie
- die Möglichkeit (auch für Kleinstfirmen), über das Internet weit entfernte Märkte kostengünstig zu erreichen [4].

Globalisierung der wirtschaftlichen und politischen Welten führt somit zu Annäherungen zwischen Staaten und Unternehmen und wird begleitet von Internationalisierungstendenzen. Sie erfordert die Erweiterung der Konkurrenzfähigkeit der Produkte von den inländischen zu den globalen Märkten. Infolgedessen werden die Unternehmen konfrontiert mit einem steigenden Preisdruck, der durch die zunehmende Anzahl von konkurrierenden Unternehmen getrieben wird. Gleichzeitig wird dabei versucht, die Absatzbasis durch Internationalisierung zu verbreitern.

Häufig wird die Globalisierung auch gemeinsam mit der Konsolidierung verfolgt. So übernehmen Automobilhersteller üblicherweise andere Hersteller fern von den

¹⁸ North Atlantic Treaty Organisation

Heimatmärkten, um durch die Akquisition in neuen Märkten das geringe Wachstum in den Triademärkten (**NAFTA**¹⁹-Staaten, Westeuropa, Japan) zu kompensieren. Ihr Wachstum verdanken daher deutsche Hersteller der Auslandsproduktion von über 40 Prozent aller Fahrzeuge. Der Wachstumsrate von 13,5 Prozent beim Außenumsatz steht dementsprechend nur ein Wert von 10 Prozent beim Inlandsumsatz gegenüber. Auch für Zulieferer wird damit die Ausrichtung auf eine globale Wertschöpfung immer notwendiger, was beispielsweise durch zusätzliche Produktionsstätten, Prozessoptimierungen und Logistikketten (Supply Chains) umgesetzt wird. Die daraus resultierenden Chancen bieten den Zulieferern erhebliche Wachstums- und Imagepotentiale:

- die Attraktivität als Arbeitgeber steigt an,
- durch Fertigungsstätten im Ausland werden oftmals Kostensituation und Lieferfähigkeit erheblich verbessert,
- die Absatzmärkte vergrößern sich, konjunkturelle Schwankungen im Heimatmarkt können besser kompensiert werden und
- das Wechselkursrisiko bei Exportgeschäften kann durch Importe teilweise verringert werden.

Wenig überraschend ist daher, dass deutsche Zulieferer bereits 45 Prozent ihres Umsatzes außerhalb Deutschlands generieren. Während Anfang der 70er Jahre nur 6 Automobilunternehmen weltweit tätig waren, belief sich ihre Zahl zwanzig Jahre später schon auf über 20 [17]. Als der dynamischste der großen Märkte von morgen wird dabei China angesehen [9].

Automobilhersteller bewegen sich in einem außerordentlich schwierigen Umfeld, in dem sie versuchen, den Druck auch über das eigene Druckniveau hinaus an die Zulieferer weiterzugeben. So lässt in den Geschäftsbeziehungen zwischen Automobilherstellern und -zulieferern die Betrachtung der Erzeugerpreise im Zeitraum von 1995 bis 2000 gemäß der Abbildung 2.3 am Beispiel des Motorenbaus eine Verlagerung des Kostendrucks auf die Seite der Zulieferer erkennen [18].

¹⁹ North American Free Trade Agreement

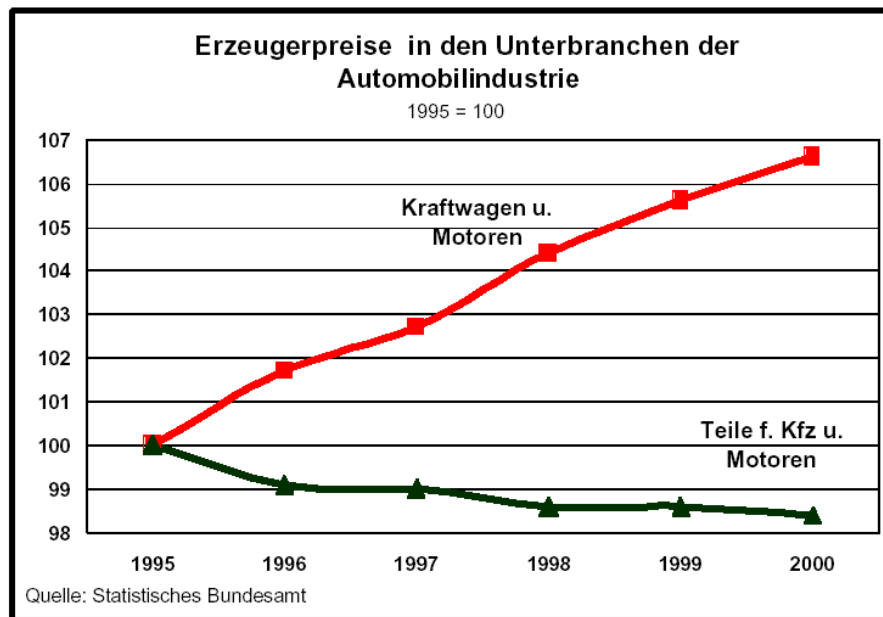


Abbildung 2.3: Veränderungen der Erzeugerpreise in der Automobilindustrie [18]

Während die Preise für Kraftwagen und Motoren um ca. 7 Prozent gestiegen sind, haben sich die einzelnen Teile dafür in dem selben Zeitraum um knapp 2 Prozent verbilligt. Sofern dabei die Ansätze zur Kostensenkung ihren Ursprung nicht in einer Effizienzsteigerung zwischen Endproduzent und Zulieferer haben, ist diese Entwicklung als äußerst problematisch anzusehen [19]. Neuen Studien zufolge resultieren aus Niedrigpreisen bei Zulieferteilen häufig schlechte Qualität und veraltete Technik [20], was sich oftmals erst Jahre später in Form von nachhaltiger Unzufriedenheit auf der Endkundenseite bemerkbar macht. Da die Wachstumsraten im internationalen Automobilmarkt für die wichtigsten Märkte in naher Zukunft bescheiden bleiben werden und der Trend zu Premium-, Klein- und Kleinstwagen sowie modischen Nischenprodukten zunehmen wird [9], wird das Überstrapazieren der Preisspirale langfristig zu Umsatzeinbußen führen. Profitable Mittelklasse-Fahrzeuge werden verdrängt und auch legislative Maßnahmen werden die Wettbewerbssituation verschärfen. Beispielsweise wurden durch eine Kfz-branchenspezifische Gruppenfreistellungsverordnung (**GVO**²⁰) die Vertriebsstrukturen innerhalb der EU vor freiem Wettbewerb und liberalem Marktgeschehen [21] geschützt. Im einzelnen ermöglicht die GVO den Herstellern eine qualitative und quantitative Absatzmittler-Selektion, d. h. das Händlernetz kann hinsichtlich Dichte und Qualitätsstandards weitgehend frei nach den Vorstellungen der Hersteller gestaltet werden.

²⁰ Gruppenfreistellungsverordnung

Innerhalb dieses selektiven Vertriebssystems wird den Händlern jeweils ein regionales Alleinvertriebsrecht innerhalb ihres Marktverantwortungsgebietes zugesichert, wobei sich der einzelne Händler im Gegenzug verpflichtet, die Neuwagen exklusiv von nur einer Marke zu vertreiben bzw. für den Vertrieb einer Zweitmarke die Genehmigung des Herstellers einzuholen [22]. Maßgebend bei der Einführung der GVO vor vielen Jahren war die Erkenntnis, dass sich die Branche ohne freien Wettbewerb besser im Sinne des Verbrauchers entwickeln würde [23]. Die Reform der GVO beendet nunmehr die Ausklammerung der Automobilindustrie von wesentlichen Teilen des Kartellrechts und macht damit das Automobil zu einem gewöhnlichen Konsumgut für die Automobilkunden, die u. a. auch in Deutschland außerordentlich anspruchsvoll geworden sind [9]. Ihren Wünschen wird durch immer mehr Nischenmodelle und Sonderausstattungs-elemente Rechnung getragen.

Begleitet werden diese Entwicklungen durch künftig marktbestimmende und technisch immer aufwendigere Ausstattungsmerkmale wie Mobile Interaktion, Telematik und den Einsatz von zusätzlicher Elektronik und Software im Fahrgastraum und außerhalb davon. Die in der Automobilindustrie im Rahmen von Modelländerungen (Modellpflegen („Facelifts“), Nachfolgermodelle) angestrebten Preiserhöhungen werden demgegenüber jedoch marktseitig immer schwieriger durchsetzbar sein, so dass auf der Herstellerseite der für die Modelländerungen betriebene Aufwand über die Zulieferer sowie über Produktivitätssteigerungen finanziert werden muss. Steigerungen in den Lohn- und Gehaltsentwicklungen, beispielsweise in Deutschland von 1999 bis 2001 um etwa 3 Prozent, führen dabei jedoch zu einem gegenläufigen preistreibenden Effekt [24].

Ein weiteres Problem ist in den letzten Jahren durch die zunehmende Produktionsausweitung in den Fabriken einerseits und die Sättigung auf den wichtigsten Absatzmärkten andererseits entstanden. So wurden erhebliche Überkapazitäten in der Automobilproduktion aufgebaut, die für Europa auf etwa 25 Prozent beziffert werden [9]. Weltweit wird gegenwärtig eine Lücke zwischen den theoretisch produzierbaren Fahrzeugen und der realen Nachfrage in der Größenordnung von rund acht Millionen Einheiten genannt [14]. Für das Jahr 2005 werden die Überkapazitäten auf 40% des dann bestehenden Produktionspotentials [25] geschätzt. Der globale Wettbewerb führt außerdem dazu, dass u. a. Fahrzeuge aus Japan und Korea für europäische Kunden interessant werden und der Verdrängungswettbewerb, der sich massiv auf der Handelsebene auswirkt, jenseits der Triade zunimmt. Beschleunigt werden diese Entwicklungen durch eine erhöhte Preistransparenz durch Internet und Euro sowie im europäischen Raum durch die erwähnte Reform der GVO. Damit ist die wichtigste Auswirkung der Globalisierung darin zu sehen,

dass sie für Unternehmen sowohl mehr Gefahren als auch mehr Möglichkeiten mit sich bringt [26].

2.3 Produkt- und Prozessinnovation

Die Produktentstehung wird in der Literatur neben der Auftragsabwicklung als der für den Erfolg eines Unternehmens bedeutendste Teilprozess betrachtet [27]. Neben den Produkten selbst rückt allerdings zunehmend die Bedeutung der zugehörigen Prozesse in den Vordergrund. Im Rahmen des Qualitätsmanagements und der Normen DIN **ISO**²¹ 9000ff, die in der Industrie durchgängig angewendet werden und die eine Dokumentation der im Unternehmen ablaufenden Prozesse fordern, sowie in den Ansätzen des **TQM**²², lassen sich analoge Aussagen erkennen [28]. Erfolg und Wettbewerbsfähigkeit eines Unternehmens werden daher künftig maßgeblich vom Reifegrad der eingesetzten Prozessstruktur beeinflusst werden.

Bei einer Zusammenarbeit mehrerer Unternehmen mit jeweils eigenständigen Aufbauorganisationen, wie dies innerhalb der Hersteller-Zulieferer-Beziehungen in der Regel der Fall ist, scheint dabei die primäre Betrachtung der durchzuführenden Prozesse über die unterschiedlichen Organisationsformen hinweg die einzige Möglichkeit zu sein, die meist sehr komplexen Vorgänge zu koordinieren und zu kontrollieren. Dies lässt sich dadurch begründen, dass sich der Prozessverlauf unter Vernachlässigung der Aufbauorganisation nur an den auszuführenden Tätigkeiten orientiert und die für den Prozess verantwortlichen Stellen direkt ansteuert. Ein Weg entlang der Instanzen, wie er durch gegebene Aufbauorganisationen erforderlich wäre, wird dadurch vermieden, was zur Reduzierung des Aufwands, der Zeit und der Kosten führt. Die Abbildung 2.4 verdeutlicht anhand eines Prozesspfades, wie eine prozessorientierte Vorgehensweise nur die jeweils für den Prozess relevanten Stellen bei Zulieferern und Herstellern ansteuert, wodurch Schnittstellenverluste und Überlastungen der Subsystemspitzen, die durch Vorgehensweisen entlang der hierarchischen Ordnung entstehen würden, vermieden werden [29].

²¹ International Organization for Standardisation

²² Total Quality Management

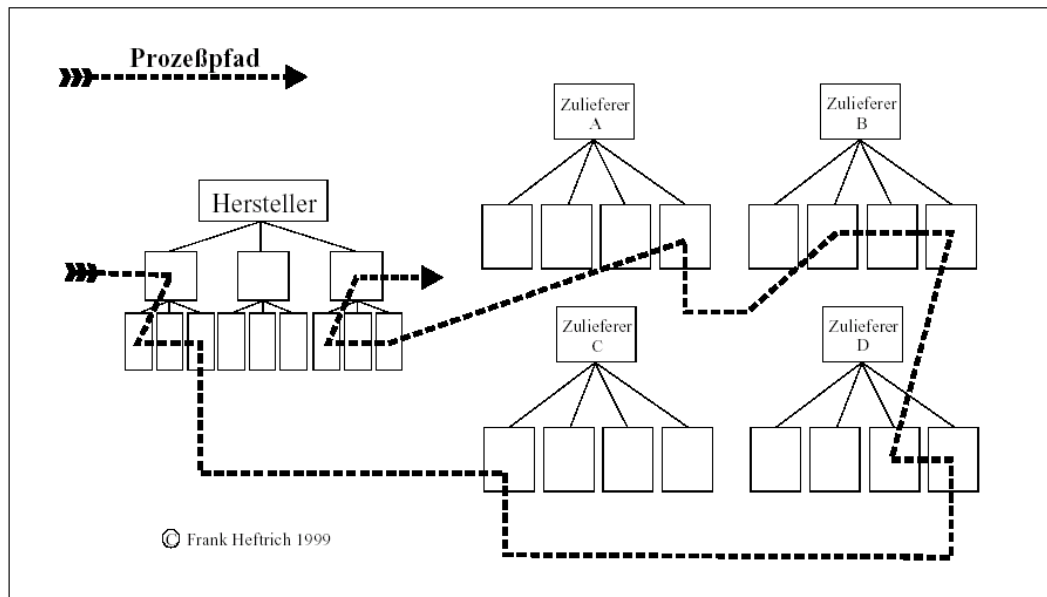


Abbildung 2.4: Prozesspfad bei Hersteller-Lieferanten-Zusammenarbeiten [30]

Prozessorientierte Ansätze stellen somit die Prozessmodelle der Zukunft dar, werden allerdings weder auf der Seite der Automobilhersteller noch auf der Seite der Zulieferer durchgängig umgesetzt. Prozessqualität und Geschwindigkeit besitzen jedoch neben der strategischen Unternehmensausrichtung eine zentrale Bedeutung, da in der Automobilindustrie Produkte für ein halbes Jahrzehnt und länger im Voraus geplant werden [9]. Wie wichtig es ist, dabei die aktuellen Marktentwicklungen nicht aus den Augen zu verlieren, zeigt das Beispiel der Volkswagen AG: während beispielsweise zu Beginn des Jahres 2003 Hersteller wie Mercedes-Benz im hart umkämpften amerikanischen Markt den Pkw-Absatz steigern konnten, musste Volkswagen einen zweistelligen Absatzrückgang hinnehmen [31]. Neben Preis und Qualität ist deshalb auch der Zeitpunkt der Markteinführung von Produkten erfolgsentscheidend [32, 33]. Dadurch wird der Faktor Zeit zu einem bedeutsamen Wettbewerbsinstrument [4].

Die parallele Verkürzung von Produktlebenszyklen [34] führt zu einem Aufbau einer neuen Form des Innovationsdrucks, und Produkt- und Prozessinnovationen werden zeitgleich immer schneller von den Konkurrenten kopiert [35]. Der Produktlebenszyklus wird zwar bei einer weiteren Verkürzung der Produktentwicklungszyklen wohl dennoch auch künftig noch im Bereich zwischen sechs bis acht Jahren liegen [9], allerdings ist mit zunehmendem technischen Fortschritt auch ein Ansteigen der Komplexität der Produkte erkennbar [36, 37]. Dadurch nimmt neben der Beschleunigung der Produktentwicklung auch der Produktumfang zu. Als Folge hiervon steigen die Amortisationszeiten für entwickelte und gefertigte Produkte

an, wodurch sich das Zeitfenster der Unternehmen, ihre Produkte in die Gewinnzone zu führen, zunehmend verengt [8].

Neben den Prozessen bei der Produktherstellung selbst werden in der Automobilindustrie auch Veränderungen bei den logistischen Prozessabläufen erforderlich [9]. Ausgehend von Nordamerika gewinnt ebenfalls in Europa beispielsweise der Lieferbarkeitszeitraum von der Bestellung bis zur Auslieferung eines Automobils an Bedeutung. So entwickelt der Hersteller Ford eine Vision des „14-Tage-Autos“, die innerhalb von 14 Tagen nach Bestelleingang eines Automobils dessen Auslieferung an den Kunden ermöglichen soll [3]. Angestrebt wird dies durch die seitens der Automobilhersteller geforderte und bereits gelebte Modularisierung und Systembauweise.

Bei Modullieferanten entsteht somit die Notwendigkeit, zahlreiche Komponenten verschiedener Unterlieferanten zu einem Gesamtmodul zusammenzusetzen und das fertige Produkt reibungslos in schmalen Zeitfenstern nach Verfahren wie JIT und JIS dem Prozess der Hersteller zuzuführen. Hierfür werden beispielsweise Zulieferparks in der Nähe der Hersteller oder eigene Logistikzentren, in denen bereits montierte Teilegruppen sequenziert werden, eingesetzt. Die betreffenden Logistikprozesse erfordern dabei eine hohe Integrationsfähigkeit untereinander, um von der Produktentwicklung über die Vorserie bis zu den Serienabrufen alle Prozessschritte abbilden zu können. Die Zulieferer übernehmen dabei vermehrt Verantwortung für die ganze Wertschöpfungskette von der Entwicklung über die Fertigung bis hin zur Supply Chain. Kriterien wie schnelle Umsetzung zur Serienreife sowie ein steigender Qualitätsanspruch der Hersteller sind deshalb ebenso zu erfüllen wie eine prozesssichere Fertigung und Anlieferung.

Vor diesem Hintergrund wird die Bedeutung von fachlichen und geografischen Kooperationen deutlich. Es sind nicht mehr nur die eigenen Ressourcen, die über den Erfolg eines Unternehmens entscheiden, sondern vielmehr die Fähigkeit, externe Ressourcen zu erschließen und in den Unternehmensprozess zu integrieren, Schnittstellen zu optimieren und die Beziehungen zu anderen Organisationen zu koordinieren. Der intraorganisatorische Optimierungsprozess hat sich in eine interorganisatorische Gesamtprozessoptimierung gewandelt [30]. Im Rahmen dieses interorganisatorischen Gesamtprozesses gilt es, über die strukturelle und prozessuale Verzahnung der Organisationsstrukturen, der Gestaltung der Informationssysteme, Unternehmenspolitik, Simultaneous Engineering-Strategie und Lernprozesse eine Veränderung der Kooperations- und Koordinationsmechanismen zu erreichen [30].

2.4 Konzentration und Konsolidierung

In kaum einer anderen als der Automobilindustrie sind Konzentrations- und Konsolidierungstendenzen offensichtlicher. Gab es in den 70er Jahren weltweit etwa 60 unabhängige Hersteller, wurden daraus durch internationale Fusionen, Aufkäufe und Zusammenschlüsse bei kontinuierlich gestiegenen Produktionszahlen Mitte 1999 nur noch 16, deren Zahl weiter gesunken ist bis auf heute 13 und sich nach Prognosen bis 2010 durch punktuelle Beteiligungen und Joint Ventures auf 5-10 Grosskonzerne reduzieren wird [9, 14, 38]. Durch diese so genannte Oligopolisierung der Kraftfahrzeugbranche werden einerseits Synergien und Skaleneffekte²³ genutzt und andererseits die eigene Marktbedeutung vergrößert. Zusätzliche Impulse werden aus den sich überlappenden Einkaufsorganisationen der konsolidierten Unternehmen generiert, die mehr Transparenz und gezieltere Preisvergleiche erhalten. Für Zulieferer führt diese Entwicklung zu einer Verringerung der Kundenanzahl. Selbst bei mengenmäßig konstanten Stückzahlen wird mit steigenden Auftragsvolumina und sinkender Auftragszahl die Wettbewerbsverschärfung vorangetrieben, denn Auftragsverluste werden immer schwerer durch Alternativen kompensierbar.

Das in der Automobilindustrie häufig zu beobachtende Outsourcing führt zu einer Neugestaltung der Wertschöpfungskette bei der Automobilherstellung. Ziel ist dabei letztlich die Konzentration auf die Kernkompetenzen, die bei den Herstellern (OEMs) derzeit die Bereiche Design und Markenführung sowie Verkauf und Service und in weiten Teilen das Engineering darstellen. Zunehmende Technisierung und Komplexität, wie beispielsweise der verstärkte Einsatz von Elektronik und Software in den Fahrzeugen, und die abnehmende Fertigungstiefe auf der Seite der Automobilhersteller führen jedoch dazu, dass eine Verlagerung zahlreicher elementarer Risiken aus Entwicklung, Design und Produkthaftung sowie Bestandhaltung und Logistik hin zu den Zulieferern stattfindet. Lag der Fertigungsanteil der OEMs im Jahr 1995 noch bei etwa 40 Prozent, so wird dieser im Jahr 2005 voraussichtlich nur noch 25 Prozent betragen [9], fahrzeugspezifisch u. U. sogar noch weniger, die Abnahme der Fertigungstiefe auf Herstellerseite wird von 71% in 1981 auf voraussichtlich 59% in 2005 prognostiziert [39]. Tendenziell findet die Verringerung der Fertigungstiefe nicht nur bei den OEMs statt, sondern bereits auf der Seite der großen Zulieferer. Die Brose Fahrzeugteile GmbH bezieht beispielsweise 60 Prozent ihrer Endprodukte von außen. Dadurch sollen kapitalintensive Produktionsschritte aus der

²³ auch: Degressionseffekt: sinkende Kosten pro Einheit bei Erhöhung der Einheitenanzahl

eigenen Fertigung ausgelagert werden, was sich durch ehrgeizige **ROCE**²⁴-Ziele v. a. auf der Seite der Hersteller gezielt bilanziell auswirkt.

Weil Übernahmen und Fusionen aufgrund der Größe der verbleibenden Hersteller zunehmend unpraktikabel geworden sind, weichen die Unternehmen immer mehr auf verschiedene Formen der Zusammenarbeit aus, um über eine Erhöhung der Ausbringungsmenge in den Genuss von Skalenvorteilen zu gelangen [40, 41]. Während gegenüber ihren Kunden die Unverwechselbarkeit des individuellen Fahrzeugs propagiert wird, bauen die Hersteller tatsächlich jedoch zahlreiche verschiedene Fahrzeugtypen auf gleichen Plattformen und damit immer baugleichen Produkten auf [42]. Anschauliches Beispiel hierfür sind die Firmen Nissan und Renault. So soll die Anzahl der von beiden Unternehmen verwendeten Plattformen von derzeit 33 (Nissan: 26; Renault: 7) auf 10 und das Motorsortiment von 27 (Nissan: 20; Renault: 7) auf 8 reduziert werden. Die Jahresproduktion wird in beiden Bereichen deutlich über 500.000 Einheiten liegen [43].

Diese Plattformstrategie beinhaltet die Vorteile, die sich aus Synergie- und Skaleneffekten ergeben, sowie Modellvielfalt bei bezahlbaren Produkten (ohne Modularisierung wären Nischenfahrzeuge unbezahlbar) und die Verkürzung der Zeit bis zur Markteinführung (time to market), wenngleich eine Profilierung und Differenzierung einer Automobilmarke von der Konkurrenz am Markt dadurch immer schwieriger wird [14]. Auch Plattformkonzepte bedeuten für Zulieferer einerseits, dass größere, aber härter umkämpfte Auftragsvolumina zu vergeben sind. Die Anforderungen an Entwicklung, Fertigung und Logistik sind dabei enorm. Andererseits steigt bei modellabhängigen Komponenten zwar die Auftragszahl, es reduzieren sich jedoch in gleichem Maße die Volumina je Auftrag.

Parallel zur Modularisierung ist die Veränderung der Fahrzeugproduktion hin zur Systembauweise offensichtlich. Systeme stellen dabei technologisch und funktional zusammenhängende Bauteile an unterschiedlichen Einbauorten dar, während Module verschiedene Funktionen an einem Einbauort zusammenführen. Immer mehr Komplettsysteme werden bei den Zulieferern bestellt [37], Beispiele hierfür sind Schiebedach, Cockpit- oder Frontendmodule. Die Gründe der OEMs liegen dabei in Einsparungen durch Verringerung der Komplexität, höhere Flexibilität in der Fertigung (fraktale und digitale Fabrik) oder marktgerechter Modellvielfalt. Die Modul- bzw. Systemlieferanten arbeiten in enger Kooperation mit dem Hersteller und liefern diesem just-in-time vormontierte, qualitäts- und funktionsgeprüfte einbaufertige Module und Systeme

²⁴ Return on Capital Employed

[44]. Bereits im Jahr 1994 wurde schon vorausgesagt, dass der Anteil langfristiger Partnerschaften zwischen Herstellern und Zulieferern von 20% (Stand 1993) auf etwa 50% der Gesamtauftragssumme der Herstellerunternehmen ansteigen wird [45].

Die Kooperationskompetenz, d. h. die Fähigkeit, Kooperationen einzugehen, wird daher als eine der wichtigsten strategischen Faktoren für erfolgreiche Unternehmen angesehen [7, 36]. Für die überwiegend mittelständisch geprägte Branche der Automobilzulieferer ergeben sich vielversprechende Chancen, die in der Kooperation liegen. Insbesondere bei den Faktoren Investitionsmöglichkeiten, Erlangung von Systemkompetenz, Transfer von Erfahrungswissen, Kostensenkung durch koordinierte Entwicklungsarbeit und geringere Gemeinkosten, Zugang zu neuen Kunden und Märkten, Internationale Marktpräsenz ohne direkte Auslandsinvestitionen, Abwicklung von Systemaufträgen, Kapazitätsauslastung und Ausnutzung von Synergieeffekten liegen große Potenziale in Kooperationen [46].

Anhand der bereits heute existierenden Zulieferparks an Montagewerken ist dabei jedoch die Rollen- und Risikoverschiebung zwischen OEM und Zulieferer erkennbar. Es sind weniger Zulieferer vom Typ „verlängerte Werkbank“, die die Zukunft prägen werden, als vielmehr solche, die sich hin zum Entwicklungs- und Designpartner entwickeln und durch Technologie- und Innovationsführerschaft ihre Position stärken.

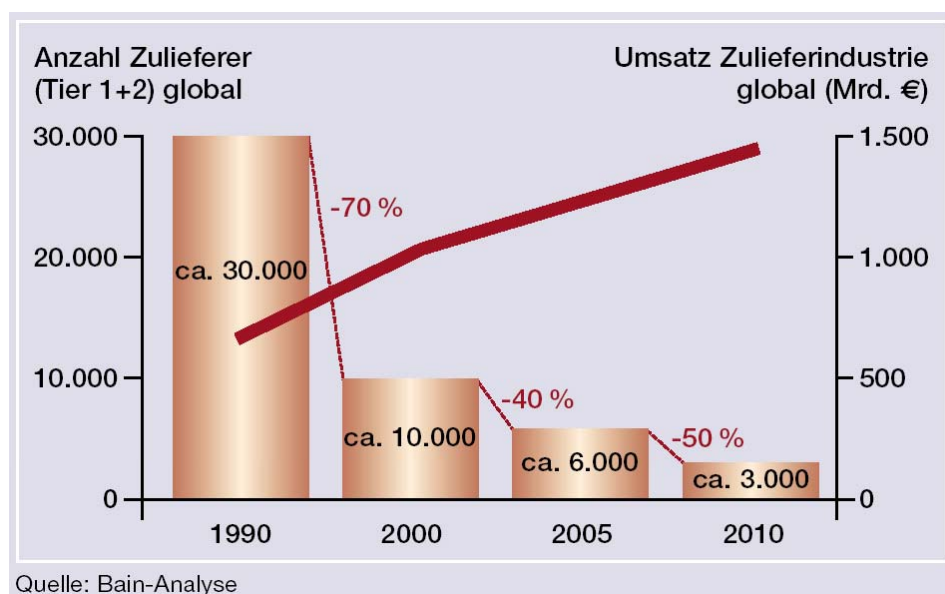


Abbildung 2.5: Veränderung von Anzahl und Umsatz der Zulieferer [9]

Diese werden die sich hieraus ergebenden Chancen nutzen, den Umsatz der übrigen Zulieferer übernehmen, verstärkt zusätzlichen Umsatz von den Herstellern abschöpfen und selbst Größenvorteile nutzen und die Effizienz ihrer Fertigungsprozesse steigern (vgl.

Abbildung 2.5). So werden von rund 30.000 Tier-1²⁵- und Tier-2²⁶-Zulieferern im Jahr 1990 nur etwa 3.000 im Jahr 2010 verbleiben. Für diese beträgt das prognostizierte Umsatzwachstum für die kommende Dekade selbst bei konservativem Szenario zwischen 30 und 40 Prozent. Wie bereits beschrieben wird das Wachstum dabei nicht durch steigenden Absatz in den Triademärkten generiert, sondern vielmehr durch weiter rückläufige Fertigungstiefe der Automobilhersteller zugunsten der Zulieferer, Aufwertung der Ausstattung und technologische Innovationen, z. B. Brennstoffzellen. Auch daran lassen sich auf Herstellerebene die Konzentrations- und Konsolidierungstendenzen erkennen.

2.5 Customer Relation

Der wirtschaftliche Erfolg in den westlichen Industrienationen und der damit verbundene gesellschaftliche Wandel hat beim Menschen das Bedürfnis nach freier Entfaltung und Individualisierung geweckt. Deshalb werden kundenspezifische Einzelleistungen und Produkte verstärkt nachgefragt [47]. Die Sättigung der Märkte unterstützt dabei die starke Kundenposition, was zur Wandlung vom Anbieter- zum Käufermarkt führt. Der Konsument und seine Wünsche stehen künftig im Mittelpunkt der geschäftlichen Aktivitäten der Unternehmen [48]. Während in der Automobilindustrie bisher von einem eher rationalen Informations- und Kaufentscheidungsprozess ausgegangen wurde, zeichnen sich in den letzten Jahren, ähnlich wie in anderen Konsum- und Gebrauchsgütermärkten [50], kundenseitig zahlreiche Veränderungen ab [51]:

- Zunehmender Informationsgrad der Käufer: Potenzielle Käufer überblicken den Markt und sind über die angebotenen Alternativen orientiert. Der kostenbewusste Konsument kann aufgrund der vielfältigen Informationsmöglichkeiten einfach zwischen verschiedenen Anbietern vergleichen und aus den Angeboten das für ihn geeignetste auswählen [52].
- Abnehmende Markentreue: Kraftfahrzeugkäufer fühlen sich immer weniger einem bestimmten Hersteller verbunden („Variety Seeking“ [53], d. h. der immer wiederkehrende und bewusst angestrebte Wechsel von Produkt, Marke und

²⁵ Tier-1 oder First Tier Lieferanten beliefern direkt die Hersteller [49]

²⁶ Tier-2 oder Second Tier Lieferanten beliefern die First Tier Lieferanten der Hersteller

Einkaufsstätte).

- Stärkeres Preisbewusstsein: Preisdifferenzen zu anderen Herstellern werden hinterfragt und nicht mehr unbegründet hingenommen.
- Hohes Qualitätsbewusstsein: Auch im unteren Marktsegment sind Kunden nicht mehr bereit, im Interesse niedriger Preise auf eine hohe Qualität zu verzichten.

Ein breiter Erfahrungsschatz, impulsives Kaufverhalten, Erlebnisorientierung sowie großes Bedürfnis nach individuellem und für Dritte erkennbarem Lebensstil sind weitere charakteristische Eigenschaften [54]. Für die Automobilindustrie ergeben sich aus diesen Entwicklungen Veränderungen, die sich vor allem auf die Wahl der Informations- und Einkaufsstätten auswirken: während Convenience-orientierte Kunden auch künftig mehrheitlich den Kauf beim Händler präferieren, ist es für den preisorientierten Smart-Shopper immer realistischer, das Fahrzeug im Internet oder im Supermarkt zu kaufen [55]. Des Weiteren verhalten sich die Kunden multioptional [56], und sie setzen ihren Gesamtkonsum aus verschiedenen Mischungen beliebig und zum Teil zufällig zusammen. Entsprechend werden Kunden zwischen Kaufsituationen, in denen sie ihre Ansprüche umfassend formulieren können und wenig Beratung benötigen, und Situationen, in denen sie auf Betreuung und Service angewiesen sind, unterscheiden [57].

Insbesondere zwingt die Entwicklung von Verkäufer- zu Käufermärkten die Unternehmen dazu, immer speziellere, auf den einzelnen Kunden abgestimmte Lösungen anzubieten. Der damit einhergehende Trend vom Massen- zum Individualprodukt führt in der Produktion zu kleinen Losgrößen. Von diesen neuen Gegebenheiten ist die gesamte Logistikkette betroffen, die sich nun an den Kundenanforderungen orientieren muss. Hier wird auch vom Wandel der Märkte vom push- zum pull-Modell gesprochen, das durch eine kunden-zentrische Sichtweise charakterisiert ist und durch Schlagworte wie „Losgröße 1“ oder auch „Mass Customization“ sowie steigende Variantenzahlen beschrieben wird [58, 59]. Den produktionstechnisch bedingten Bemühungen um Vereinheitlichung steht insofern ein deutlicher Trend zur Individualisierung im Bereich der Nachfrage gegenüber, der sich im Zuge des Wertewandels vor allem in den westlichen Ländern ergeben hat. Autokäufer sind daher zunehmend unzufrieden mit Einheitslösungen. Vielmehr ist die Notwendigkeit, dass das Automobil mit den persönlichen Anforderungen und Wünschen im Einklang steht, zu einem wesentlichen Kriterium der Kaufentscheidung geworden.

Unter der Berücksichtigung der Möglichkeiten der neuen Informations- und Kommunikationstechnologien an der Schnittstelle zum Kunden lassen sich drei zentrale Einsatzmöglichkeiten für die Automobilbranche unterscheiden [60]: Verkaufsanbahnung, Auktionen und Transaktionen. Diesbezüglich sind Interaktivität, Multimedialität, interne und externe Vernetzung, Such- und Selektionsfunktionen, Möglichkeiten der permanenten Aktualisierung, Archivzugriff und schnelle Verfügbarkeit nur einige Schlagworte, unter welchen die Möglichkeiten des Internet diskutiert werden [61].

Endkundenseitig sind im Marketing via Internet auf dem deutschen Markt momentan vor allem Neu- und Gebrauchtwagenbörsen sowie Websites von Automobilherstellern und –händlern von großer Bedeutung [60]. Aus aktuellen Prognosen geht hervor, dass im Jahr 2010 der Neuwagenvertrieb über das Internet für Premiummarken 10% und für Volumenmarken 20% des gesamten Marktes betragen wird, während der Marktanteil des traditionellen Vertragshandels von heute 80% auf 60% absinken wird [62].

Auch in den Beziehungen zwischen OEM und Zulieferer lassen sich auf den Ebenen Produktion und Zusammenarbeit charakteristische Entwicklungsstufen erkennen (vgl. Abbildung 2.6).

Era	Mass Manufacturing <i>Early automotive years</i>	Vertical Integration <i>1950s through 1970s</i>	Core Competency <i>1980s through 1990s</i>	Collaborative Community <i>2nd automotive century</i>
Industry Characteristics	<ul style="list-style-type: none"> ▪ New production efficiencies ▪ Stable OEM & supplier relationships 	<ul style="list-style-type: none"> ▪ Recognition of collaboration needs ▪ Economies of scale ▪ Capability focused consolidation 	<ul style="list-style-type: none"> ▪ Competitive pressures ▪ Globalisation ▪ Excess capacity ▪ Lost economies of scale ▪ Alignment of resources 	<ul style="list-style-type: none"> ▪ Harshening Competition ▪ Asset redundancies ▪ Value destruction ▪ Increasing consumer expectations ▪ Innovation needed
Collaboration Characteristics	<ul style="list-style-type: none"> ▪ Defined by manufacturing capability assessments ▪ Limited to communication of production needs 	<ul style="list-style-type: none"> ▪ „Internal“ collaboration as vertical integrat. ▪ Collaboration as consolidation rather than co-operation ▪ Collaboration around individual product programmes 	<ul style="list-style-type: none"> ▪ Redefinition of Collaborative Eng. ▪ Enhanced information exchange in platform planning, concept, design & manufacturing ▪ New technologies: CAD/CAM 	<ul style="list-style-type: none"> ▪ New Business Model required ▪ Collaborative Communities ▪ Consumer Centricity ▪ Redefined Value Distribution

Abbildung 2.6: Entwicklungsstufen zwischen OEM und Zulieferer [63]

Seit dem Beginn der Massenproduktion in der Automobilindustrie führten die Marktanforderungen über zwei Epochen in den Zeiträumen von 1950 bis heute zu einem zunehmenden, sich verschärfenden Wettbewerb, Kapazitätsüberhängen, steigenden

Kundenerwartungen und Innovationsdruck, was in den letzten Jahren eine enorme Wertevernichtung an den internationalen Aktienmärkten auslöste und damit den Handlungsdruck erheblich verstärkt hat.

Stand noch vor einigen Jahrzehnten die Teilefertigung im Fokus der Rationalisierung, so gewann in den letzten 15 Jahren die Montageautomatisierung immer mehr an Bedeutung [64]. Mit fortschreitender Globalisierung treten die Unternehmen immer weiter in einen internationalen Konkurrenzkampf [65]. Um dem dadurch verursachten Kostendruck zu begegnen, ist es erforderlich, sich von den Mitbewerbern zu differenzieren. Dies kann unter Berücksichtigung des Kundenwunsches nach individuellen Produkten durch Verbreiterung der Sortimente und Erhöhung der Variantenvielfalt geschehen [66]. Innovative Erzeugnisse sind schnell, in optimaler Qualität, zu attraktiven Preisen und in ausreichenden Mengen auf den Markt zu bringen. Die logistischen Qualitätsmerkmale eines Unternehmens werden immer wichtiger.

In einer Untersuchung zu den Kundenanforderungen an die Distributionslogistik gaben die befragten Unternehmen als wichtigsten Punkt die Termintreue an [67]. Auch Wiendahl schreibt einer kurzen Lieferzeit und der verlässlichen Einhaltung der zugesagten Termine eine große Bedeutung für den Unternehmenserfolg zu [68]. Durch die stetige Verkürzung des Produktlebenszyklus ist die Reduzierung der „time to market“ als Wettbewerbsfaktor unumgänglich [69]. So wurde z. B. die Zeit von der Konzeption bis zur Markteinführung eines Pkw von 48 Monaten auf 24 Monate verkürzt [70]. Kurze Innovationszyklen erfordern, dass sich die Produktionsanlagen den geänderten Gegebenheiten schnell anpassen müssen [71]. Dies gilt ebenso für die direkt und indirekt angebundenen Systeme.

Es sind in Anlehnung an die Abbildung 2.6 neue Formen der Zusammenarbeit, mit denen diesen Herausforderungen zu begegnen sein wird. Hierbei muss das Ziel lauten, die Wertschöpfungskette neu zu gestalten und die Prozesse an den Anforderungen der Kunden auszurichten.

2.6 Transport – und Lagerlogistik

Gleichzeitig mit dem Wandel vom Anbieter- zum Käufermarkt vollzieht sich die Internationalisierung der Beschaffungs- und Absatzmärkte. Deshalb wird die Logistik immer wichtiger. Erst seit einigen Jahren werden die Rationalisierungspotentiale der inner- und

außerbetrieblichen Logistik untersucht. So liegt der Anteil der Logistikkosten am Umsatz, abhängig vom betrachteten Industriezweig, zwischen 10 Prozent (Maschinenbau) und 30 Prozent (Nahrungsmittelindustrie) [72, 73], das Kostensenkungspotential durch die Logistik wird in der Industrie mit 6 Prozent und im Handel mit 7 Prozent angegeben [74].

Ein Problem, das nicht nur in der Automobilindustrie existiert, ist die Bindung von Kapital in Form von Lagerbeständen bei Material, fremdbezogenen Teilen, Roh-, Hilfs- und Betriebsstoffen sowie Halb- und Fertigware. Deshalb zählen die daraus entstehenden Kosten zu den wichtigsten Ansatzpunkten zur Verbesserung des Unternehmensergebnisses und damit zur Sicherstellung der Wettbewerbsfähigkeit [75]. Hohe Flexibilität und Lieferbereitschaft bei möglichst niedrigen Beständen stellen daher die im Vordergrund stehenden Ziele dar.

Untersuchungen haben ergeben, dass die Durchlaufzeit des Transportguts durch den Betrieb ca. 75 Prozent Wartezeit beinhaltet [76, 77, 78]. Der vermeidbare arbeitsablauf- und störungsbedingte Liegezeitanteil kann dabei nur durch eine Optimierung des Materialflusssystems und dessen Konfiguration reduziert werden. Ein störungsärmerer und beschleunigter Materialfluss bei geringeren Transportkosten ist die Folge [79]. Ausgehend von den Materialflussskosten in der Produktion, die, abhängig von den betrieblichen Gegebenheiten und der Beschaffenheit der Erzeugnisse, einen Anteil von 20-40 Prozent an den Fertigungskosten besitzen, werden durch die kürzere Aufenthaltszeit der Güter im Materialflusssystem die Bestände gesenkt, was die Kapitalbindungskosten der Umlaufbestände positiv beeinflusst [80].

Eine Betrachtung der Kosteneinspareffekte aus der Perspektive der Wertschöpfungszeiten führt bei einer Veränderung des Wertschöpfungsanteils von 5 auf 10 Prozent zu einer Reduzierung der Liege- und Wartezeiten von 95 auf 90 Prozent. Dadurch werden insgesamt die Durchlaufzeiten halbiert, die Lieferfähigkeit deutlich erhöht, die Material- und Halbfabrikate-Bestände um 50 Prozent reduziert (exkl. der Sicherheitsbestände) sowie die Ressourcenplanung verbessert und die Fertigungsflexibilität gesteigert [81].

Die über Bestandssenkungen hervorgerufenen Effekte der Materialkostenreduzierung zeigt anschaulich die nachstehende Abbildung 2.7, in der anhand eines Materialkostenanteils am Umsatz von 45 Prozent die erforderliche Umsatzsteigerung abgebildet ist, mit der der gleiche Gewinneffekt erzielt werden kann. Demzufolge resultiert aus einer Materialkostenreduzierung von 0,5 Prozent ein Gewinnbeitrag, der sonst nur mit einer

Umsatzsteigerung von 4 Prozent zu realisieren wäre, bei einer Materialkostenreduzierung von 2 Prozent ist bereits eine Umsatzsteigerung von 16 Prozent erforderlich.

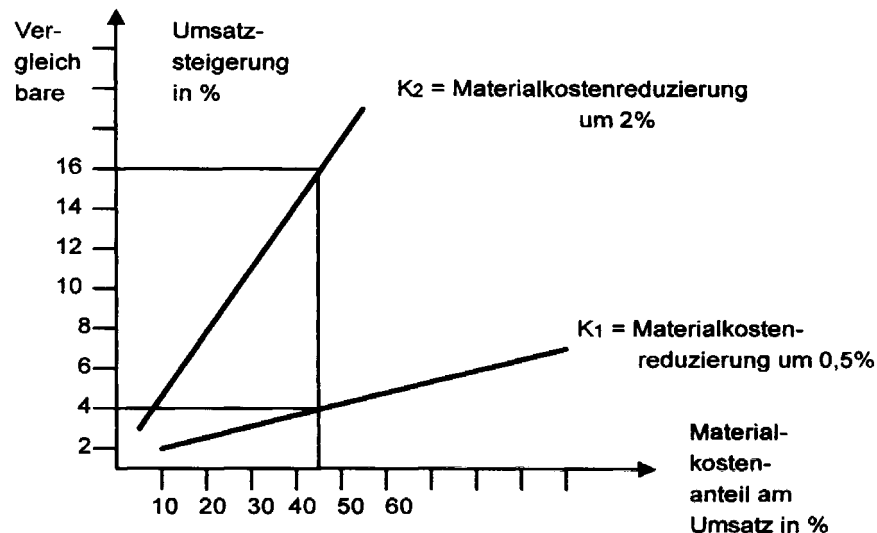


Abbildung 2.7: Gewinneffekte bei Materialkostenreduzierung und Umsatzsteigerung [81]

Andere Berechnungen ergeben bei einem Vorratsanteil von 20 Prozent des Umsatzes und einer Bestandsreduzierung um 10 Prozent eine äquivalente Umsatzsteigerung von 18 Prozent [82]. Der Reduzierung von Lagerbeständen stehen in der Praxis jedoch eine Reihe von Faktoren gegenüber, die gegenteilige Anforderungen stellen. Hierzu zählen die Umsetzung kurzer Liefertermine und hoher Termintreue, hoher Flexibilität und Lieferbereitschaft sowie geringer Einstandspreise, Beschaffungskosten und Rüstkosten. Es gilt daher, die genannten Parameter derart in ein Gleichgewicht zu bringen, dass insgesamt ein Kostenoptimum erreicht wird.

Ein in diesem Zusammenhang bekannt gewordenes Phänomen stellt beispielsweise der sogenannte Bull-Whip-Effekt (vgl. Abbildung 2.8) dar [83]. Da kundenseitig die Bedarfs- und Bestandsverläufe im einzelnen Unternehmen nicht isoliert betrachtet werden können, sondern vielmehr die gesamte logistische Kette im Fokus steht, unterliegen die Bedarfsverläufe von einem Unternehmen zum nächsten mitunter zeitlichen Verschiebungen, die sich wie ein Peitscheneffekt aufschaukeln können. Der Grund hierfür ist, dass jeder Partner einer logistischen Kette vom Produzenten bis zum Einzelhändler üblicherweise nur diejenigen Bedarfe kennt, die ihm von seinem Kunden direkt gemeldet werden [3]. Dem Aufkommen von Fehlbeständen wird mit dem Aufbau von Sicherheitsbeständen

entgegengewirkt, die in allen Lieferstufen vorgehalten werden, und so werden der Reduzierung der Lagerbestände systemimmanent Grenzen gesetzt, die letztlich das Ziel der bedarfsorientierten Disposition zu Gunsten der verbrauchsorientierten verfehlen, bei der regelmäßig entweder zu viel oder zu wenig produziert wird.

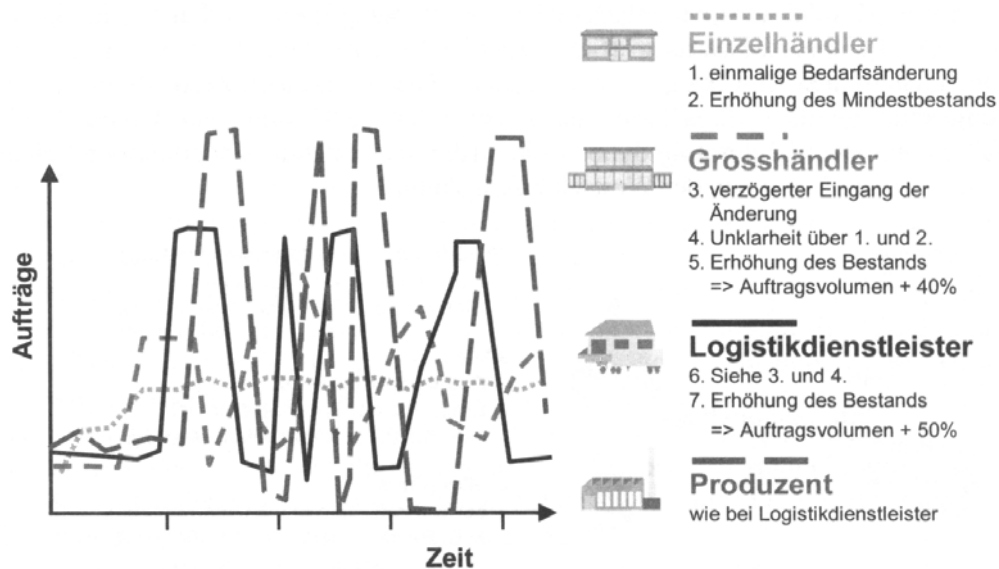


Abbildung 2.8: Bull-Whip-Effekt [83]

Detaillierte Gründe für den Bull-Whip-Effekt werden in der Nachfrageverstärkung durch Sicherheitsdenken und Losgrößen gesehen, außerdem im Burbidge-Effekt (Nachfrageverstärkung durch unterschiedliche Bestellperioden) und im Forrester-Effekt (Nachfrageverstärkung durch verzögerten Informationsfluss) [83].

Neben den internen Logistikkosten erweisen sich aus Herstellersicht auch die gewandelten Kostenstrukturen als besondere Herausforderung: nach Jahren der Neuausrichtung bei Forschungs- und Entwicklungs-Aktivitäten, der Effizienzsteigerung in der Beschaffung und des Aufbaus einer "Lean Production" werden nun in der Distribution maßgebliche Rationalisierungspotentiale identifiziert [84]. So gehen verschiedene Anbieter davon aus, dass die Distributionskosten heute ca. 30-40 % des Listenpreises ausmachen [14]. Allein dadurch wird deutlich, dass die Kosten im Bereich der Transport- und Lagerlogistik maßgeblichen Einfluss auf die Gesamtkostenstruktur ausüben.

2.7 Informations- und Kommunikationsbarrieren

Die Information wird in der Betriebswirtschaft als weiterer Produktionsfaktor neben Arbeitsleistung, Betriebsmitteln, Energie, Finanzierungsmitteln, etc. bezeichnet, Information und Kommunikation werden dabei als zentrale Bestandteile des ökonomischen Handelns angesehen [85]. Informations- und Kommunikationstechnologien tragen mittlerweile zur Vereinfachung der Koordination, zur Erhöhung der Informationsaktualität sowie zur Überwindung räumlicher Distanzen bei und helfen damit ganz wesentlich, die Möglichkeiten unternehmensübergreifender Arbeitsteilung erheblich zu verbessern [86, 87]. Die Kooperations- und Innovationsfähigkeit kann durch moderne Informationstechnologien erheblich gesteigert werden [88, 89]. Die Vorteile einer frühzeitigen und ganzheitlichen Integration aller direkt oder indirekt am Innovationsprozess beteiligten Unternehmen, Subsysteme und Mitarbeiter sind bereits seit längerem bekannt [90, 91], Grundlagenbetrachtungen, Funktionsbeschreibungen und definitorische Arbeit im Bereich Information, Informationsprozesse und –systeme sind in zahlreichen wirtschaftswissenschaftlichen Arbeiten zu finden, deren Ergebnisse auf die Zusammenarbeit von Unternehmen übertragbar sind [30].

Obwohl etwa 70 % der Wertschöpfung am Endprodukt, das in der Regel das fertige Automobil ist, bei den Zulieferbetrieben geleistet wird, sind die Informations- und Materialwege entlang der logistischen Struktur bis hin zum Automobilhersteller noch keinesfalls durchgängig. Vielmehr ist zu beobachten, dass sehr häufig proprietäre Kommunikationsverbindungen zwischen zwei Partnern bestehen [3], allerdings keine Verbindungen mehr zu vor- oder nachgelagerten Lieferanten oder Kunden dieser Partner. Die zeitverzögerte Informationsweitergabe von einem Glied der logistischen Kette zum nächsten führt zum bereits erwähnten Forrester- und schließlich zum Bull-Whip-Effekt, der bei kleinen Änderungen des Bedarfs beim Endkunden zu immer größeren Bedarfschwankungen der vorgelagerten Lieferanten führt, je weiter die logistische Kette zurückverfolgt wird. Die Automobilindustrie begegnet dem Informationsversatz bei gleichzeitig vorhandenem Lieferzwang üblicherweise mit dem Aufbau von Sicherheitsbeständen mit all den Konsequenzen, die bereits im letzten Kapitel erläutert wurden.

In der Geschäftsbeziehung zwischen Automobilhersteller und –zulieferer kommt den Schnittstellen zwischen beiden Partnern dabei eine zentrale Bedeutung zu [9]. Unter dem Schlagwort **E-Business**²⁷ werden Prozesse schrittweise entlang der gesamten

²⁷ Electronic Business

Wertschöpfungskette standardisiert, automatisiert und digitalisiert, wodurch Prozesse niedrigere Kosten aufweisen, effizienter und schneller werden. Ermöglicht wird dies durch die Vernetzung der beteiligten Partner. Im strategischen Einkauf haben Automobilhersteller beispielsweise in Form von „Reverse-Online-Auktionen“ bereits Lösungen etabliert (Supply-On, covisint), wenngleich der erhoffte Nutzen noch hinter den Erwartungen zurückbleibt. Unabhängig davon werden die nächsten Schritte die weitere Vernetzung von Hersteller und Zulieferer sein (beispielsweise auf Basis der standardisierten Kommunikationsnetzwerke **ENX**²⁸ oder **ANX**²⁹), wobei die Supply-Chain-Integration weitere Potentiale erschließen wird. Hierzu zählen die Senkung von Transaktions- und Produktkosten, erhöhte Transparenz hinsichtlich Lagerbeständen, Durchlaufzeiten und Kostenstrukturen, Zunahme an Produktivität und Qualität oder Verkürzung der time to market.

Grundsätzlich ist der effiziente Einsatz moderner Kommunikationstechnologie sowohl auf der Beschaffungs- und Entwicklungsseite als auch auf der Absatzseite für Unternehmen im Automobilzulieferbereich eine notwendige Bedingung. Hieraus erwachsen die Möglichkeiten einer frühzeitigen und engen Zusammenarbeit zwischen Kunden, Lieferanten und Entwicklungspartnern mit dem Ziel, die prozessbeteiligten Ressourcen zu koordinieren. Dennoch gestaltet sich der Einsatz unternehmensinterner oder -übergreifender IT-Systeme mitunter schwierig, was bereits bei der Einführung von **CIM**³⁰-Systemen deutlich wurde. Unter Computer Integrated Manufacturing wird die integrierte Informationsverarbeitung für betriebswirtschaftliche und technische Aufgaben eines Industriebetriebes verstanden [92]. CIM zielt dabei speziell auf folgende Felder der Wettbewerbsstrategie ab: Vertrieb, Produktion, F&E, Einkauf, Personal und Finanzierung. Es setzt sich in der Regel aus den Teilsystemen **CAE**³¹, **CAP**³², CAD, **CAM**³³, **CAQ**³⁴, BDE, **PPS**³⁵ zusammen und ist eine der funktionalen Strategie übergeordnete Konzeption, die als Querschnittsfunktion die informationsverarbeitenden Prozesse aller Anwendungsbereiche mit einbezieht.

Es gibt Bestrebungen, das CIM-Konzept durch zusätzliche Komponenten zu erweitern, indem das Rechnungswesen, die Unternehmensplanung, die Bürokommunikation und das Personalwesen innerhalb von CIM integriert werden. Dies verbessert die Möglichkeiten des operativen und strategischen Controlling. Die Unternehmensleitung verfügt so über eine

²⁸ European Automotive Network Exchange

²⁹ Advanced Network Exchange

³⁰ Computer Integrated Manufacturing

³¹ Computer Aided Engineering

³² Computer Aided Production

³³ Computer Aided Manufacturing

³⁴ Computer Aided Quality

³⁵ Produktionsplanung und -steuerung

Vielzahl von Informationen, um Entscheidungen im Rahmen der Unternehmensplanung besser zu fundieren. Jetzt wird nicht mehr von CIM, sondern von **CAI**³⁶ gesprochen [93]. Dieser Begriff scheint auch für die bisherigen CIM-Integrationsmodelle wesentlich besser geeignet zu sein, da CIM bereits wesentlich mehr enthält als nur den materiellen Umwandlungsprozess in der Fabrik. CIM begleitet den gesamten Produktentstehungsprozess, vom ersten Entwicklungsschritt über die Planung und die materielle Umwandlung bis hin zum Vertrieb des fertigen Produktes.

Arbeiten Hersteller und Zulieferer eng zusammen, wird meist von seiten der Hersteller ein einheitliches IT-System gefordert. Der Zulieferer ist gezwungen, dieses System zu installieren, bei Beziehungen zu mehreren Herstellern mit unterschiedlichen Systemen kann dies für den Zulieferer zu erheblichen Investitionskosten führen [94].

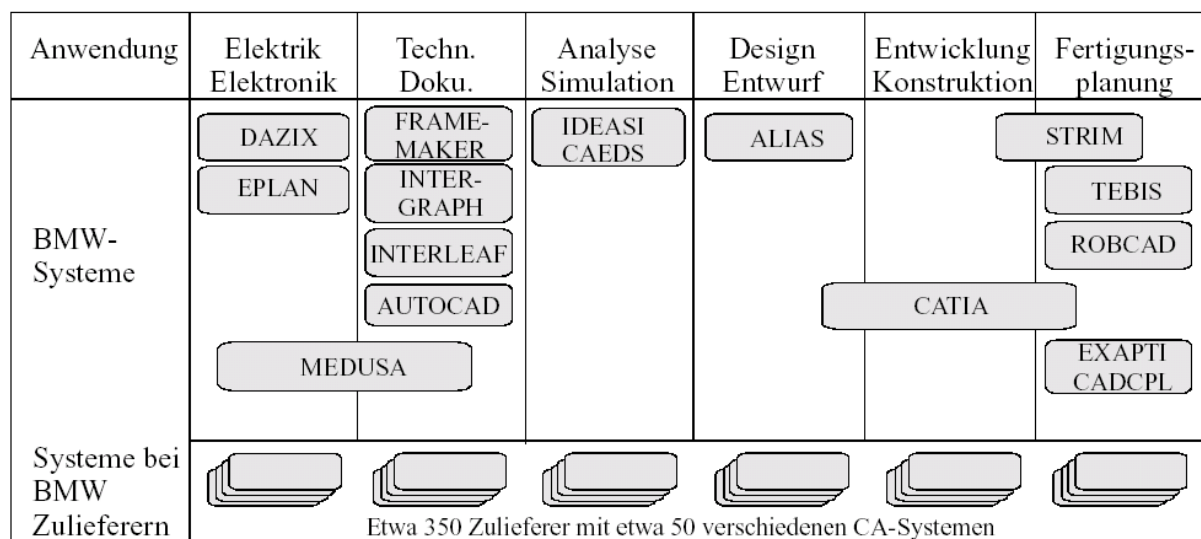


Abbildung 2.9: Variantenvielfalt der eingesetzten CA-Systeme am Beispiel BMW [30]

Die Abbildung 2.9 verdeutlicht in diesem Zusammenhang am Beispiel von BMW und seinen Zulieferern die Vielzahl an unterschiedlichen **CA**³⁷-Komponenten, die dort entlang der Wertschöpfungskette Automobil zum Einsatz kommen. Die insgesamt 350 beteiligten Zulieferer arbeiten hier für die unterschiedlichen Einsatzzwecke wie Design, Entwicklung, Analyse und Dokumentation oder für die Fertigungsplanung mit etwa 50 verschiedenen Anwendungen.

³⁶ Computer Aided Industry

³⁷ Computer Aided

Wird andererseits die Systemlandschaft eines Unternehmens entlang der Wertschöpfungskette betrachtet, so lässt sich die Existenz einer Reihe von Anwendungssystemen erkennen, die sich danach unterscheiden lassen, welche Aufgaben und Prozesse von ihnen unterstützt werden (vgl. Abbildung 2.10).

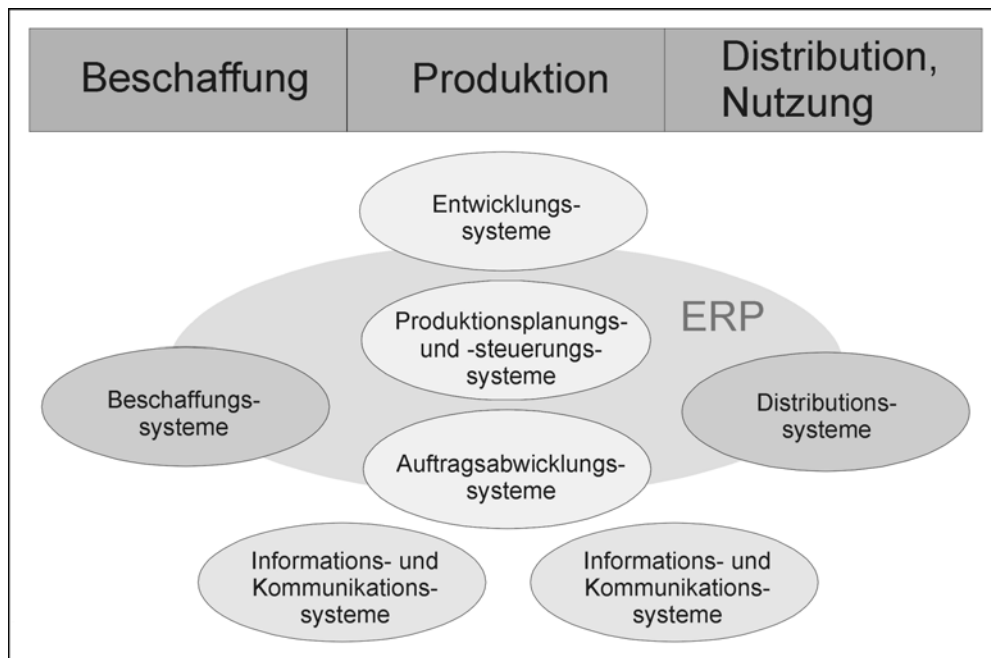


Abbildung 2.10: Anwendungssysteme in der logistischen Kette

Während Entwicklungssysteme die rechnergestützte Produktentwicklung abbilden (CAD, CAM), handelt es sich bei Beschaffungssystemen um rechnergestützte Einkaufssysteme, mit denen Prozesse wie Electronic Procurement, Lieferantenbewertungen oder Lieferabrufe durchgeführt werden. Demgegenüber stellen Distributionssysteme eine rechnergestützte Warenverteilung dar, mit der die Bereiche Warenwirtschaft bzw. Lagerverwaltung und Kommissionierung abgedeckt werden. Als Kernelemente existieren die PPS-Systeme, die sich mit der Arbeitsvorbereitung und der (mehrstufigen) Fertigungssteuerung beschäftigen, sowie die Auftragsabwicklungssysteme, die alle Schritte des Auftragsgewinnungsprozesses beinhalten (Vertriebs-/Versandabwicklung, Auftragsdurchführung (**CAS**³⁸) und **CRM**³⁹) sowie elektronischer Handel (**e-commerce**⁴⁰)). Schließlich existiert eine Reihe von Informations- und Kommunikationssystemen, die einerseits für die Identifikation (Warenerfassung,

³⁸ Computer Aided Selling

³⁹ Customer Relationship Management

⁴⁰ electronic commerce

Kennzeichnung, Leergut- und Retourenverwaltung) sowie für die Kommunikation (**EDI**⁴¹, **DFÜ**⁴²) zuständig sind [3].

Im mittelständischen Umfeld löste vor wenigen Jahren das **Y2k**⁴³-Problem eine Welle von Investitionen aus, die häufig dazu führte, dass die informationstechnische Infrastruktur in vielen Unternehmen in weiten Teilen erneuert oder erweitert wurde. So stand u. a. in dieser Zeit die Einführung, die Erneuerung oder die Standardisierung eines PPS-Systems in vielen mittelständischen Unternehmen im Vordergrund. Diese Systeme werden wie oben beschrieben zur Planung, Steuerung und Überwachung der Produktionsabläufe eingesetzt, wobei häufig Algorithmen zur Berücksichtigung von Mengen, Terminen, Kosten und Kapazitäten verwendet werden. Typischerweise handelt es sich bei den PPS-Systemen um Lösungen, die vorrangig eine werkszentrische Sicht ermöglichen und auch das nur in den bereichsspezifischen Aufgaben „Beschaffen“ und „Herstellen“ im Sinne der Versorgung bzw. Materialwirtschaft.

Über die PPS-Funktionalität hinaus wurden vielfach ERP-Systeme installiert, die Softwaremodule zur Finanzwirtschaft, Einkauf und Materialwirtschaft, Entwicklung sowie Unternehmensführung nutzen. Obwohl teilweise eine funktionale Erweiterung durch ERP-Systeme über Systemgrenzen hinaus stattfindet (vgl. Abbildung 2.10), bleibt auch wie bei den PPS-Systemen die Betrachtung auf innerbetriebliche Aufgaben begrenzt und darüber hinaus im Sinne einer anwendungssystemseitigen Sichtweise selten ganzheitlich.

Beispielhaft seien in diesem Zusammenhang die bei der Anwendung von PPS- und ERP-Systemen typischerweise auftretenden Probleme genannt:

- die Datenweitergabe an Schnittstellen zu anderen Softwaresystemen ist nur mit zusätzlichen Systemen bzw. mit zusätzlicher Software möglich; darüber hinaus weisen die verfügbaren Integrationsverfahren eine für den Mittelstand beachtliche Kostenstruktur auf. Änderungen oder Updates an einem der beteiligten Systeme führen weiterhin zu einem nachhaltigen Investitionsdruck. Verstärkt wird dieser Effekt dadurch, dass die benachbarten Organisationsbereiche des PPS-Systems wie beispielsweise CAQ, CAD oder CRM in vielen Fällen nur unzureichend berücksichtigt sind und damit zu keinem ganzheitlichen System führen. Dadurch wird die

⁴¹ Electronic Data Interchange

⁴² Datenfernübertragung

⁴³ Jahr-2000

Heterogenität in der Systemlandschaft weiter vergrößert, was zu einer Vervielfältigung der Schnittstellenproblematik führt.

- ist der Datenaustausch zwischen Softwareprodukten verschiedener Hersteller notwendig, so ist dies, wenn überhaupt, nur durch Konvertierung von Satz- oder Datenformaten möglich. Dies stellt allerdings ein generelles Problem beim Einsatz von Software dar.
- es sind nur in begrenztem Umfang Reaktionsmöglichkeiten auf Veränderungen der externen Rahmenbedingungen vorhanden. Hierzu zählen beispielsweise Konstruktionsänderungen bei bestehenden Produkten oder Neuentwicklungen, die neue Anforderungen an den Beschaffungsprozess stellen oder zu veränderten Arbeitsplänen und Produktionsabläufen führen. Durch die sukzessive Kapazitätsplanung, die auf konstanten Kapazitäten beruht, wird ein dynamisches Reagieren in der logistischen Kette zusätzlich erschwert, was noch verstärkt wird durch eine statische Auftragsfeinplanung nach Auftragserteilung. Auftragsänderungen oder –ergänzungen können nicht mehr berücksichtigt werden, was bei kritischen Teilen dazu führt, dass „auf Verdacht“ produziert wird.
- Stammdaten sind teilweise nicht gepflegt und der Aufwand zur laufenden Datenpflege ist hoch. Dies gilt beispielsweise auch für Umlaufbestände und fertigungsnahe Puffer, die zu unzureichenden Informationen bei der Bestandsführung in fertigungsnahen Bereichen führen.
- Abläufe sind im System nur mangelhaft nachvollziehbar.
- die Auftragsverfolgung ist umständlich oder unmöglich.
- bei Störungen des Systems ist auch der Gesamtablauf massiv gestört.

Bereits heute sind PPS- und ERP-Systeme, die in der Automobilindustrie eingesetzt werden, zumeist in der Lage, untereinander Daten auszutauschen. Die zugrundeliegenden Verfahren werden beispielsweise zur Übertragung von Auftrags-, Bestell- oder Lieferinformationen verwendet und übergeordnet als EDI oder DFÜ bezeichnet. Die Zielstellung ist hierbei, die Daten eines Anwendungssystems ohne zusätzliche manuelle Bearbeitung direkt an das

Zielsystem des Empfängers weiterzugeben. Gängige EDI-Verfahren sind **EDIFACT**⁴⁴ zur branchenübergreifenden Übertragung standardisierter Nachrichten über Bestellungen, Lieferungen, Rechnungen, usw., **ODETTE**⁴⁵ zur übergreifenden Datenübertragung in der Automobilindustrie (z. B. CAD-Daten) und **VDA**⁴⁶-Standard zur Übertragung von Liefer- oder Feinabrufen in der Automobilindustrie. Üblicherweise werden diese elektronischen Systemverknüpfungen mit vielen unterschiedlichen und relativ teuren Infrastrukturen geschaffen (z. B. VDA via **ISDN**⁴⁷, **X.25**⁴⁸ oder **X.31**⁴⁹). Eine Standardisierung, wie sie beispielsweise über das Internet möglich wäre, ist zwar mittels Web-EDI teilweise erkennbar, allerdings ist die verwendete Infrastruktur nach wie vor sehr heterogen. Dies stellt für mittelständische Unternehmen einen beachtlichen administrativen und finanziellen Aufwand dar, da Verbindungen mitunter kundenbezogen eigens geschaffen und aufrechterhalten werden müssen.

Im Planungsablauf selbst gibt es oft keine Durchgängigkeit der Daten. Je nach Planungsphase werden unterschiedliche, für den speziellen Anwendungsfall geeignete Programme eingesetzt, die jeweils eigene Datenformate verwenden. Selten existieren Schnittstellen zur direkten Übernahme der Daten aus vorangegangenen Planungstätigkeiten. Durch mehrfache Datenhaltung und -eingabe sind Aktualität, Vollständigkeit, Plausibilität und Konsistenz und damit die Datenqualität nicht sichergestellt. Die eingesetzten Anwendungen stellen Insellösungen ohne technische und inhaltliche Kopplung dar [95].

Als Folge der nicht durchgängigen Systemverknüpfungen wird in großem Maße die Beherrschbarkeit der Gesamtprozesse hinsichtlich der Waren- und Informationsflüsse und damit der Supply Chain erschwert, da zwangsläufig zeitliche Schwankungen in die Prozesse eingetragen werden, die kaum planbar sind und die von der Systemleistung der Partnersysteme abhängig sind. Das Problem verschärft sich dann, wenn mehrere Systeme an den Prozessen beteiligt sind, jedoch keine oder nur teilweise eine Verbindung zwischen diesen Systemen besteht. Dies ist allerdings die übliche Situation in den heutigen heterogenen IT-Systemlandschaften mittelständischer Unternehmen. Während die Kommunikation zwischen einzelnen Systemen noch über Schnittstellen realisiert ist, gibt es

⁴⁴ Electronic Data Interchange for Administration

⁴⁵ Organisation for Data Exchange by Tele Transmission in Europe

⁴⁶ Verband der deutschen Automobilindustrie

⁴⁷ Integrated Services Digital Network

⁴⁸ Kommunikationsprotokoll für den Datenaustausch

⁴⁹ Kommunikationsprotokoll für den Datenaustausch

Anwendungen, die keine Verbindung zu den an der Supply Chain beteiligten Prozesssystemen besitzen und zwischen denen daher eine Kommunikation nicht möglich ist.

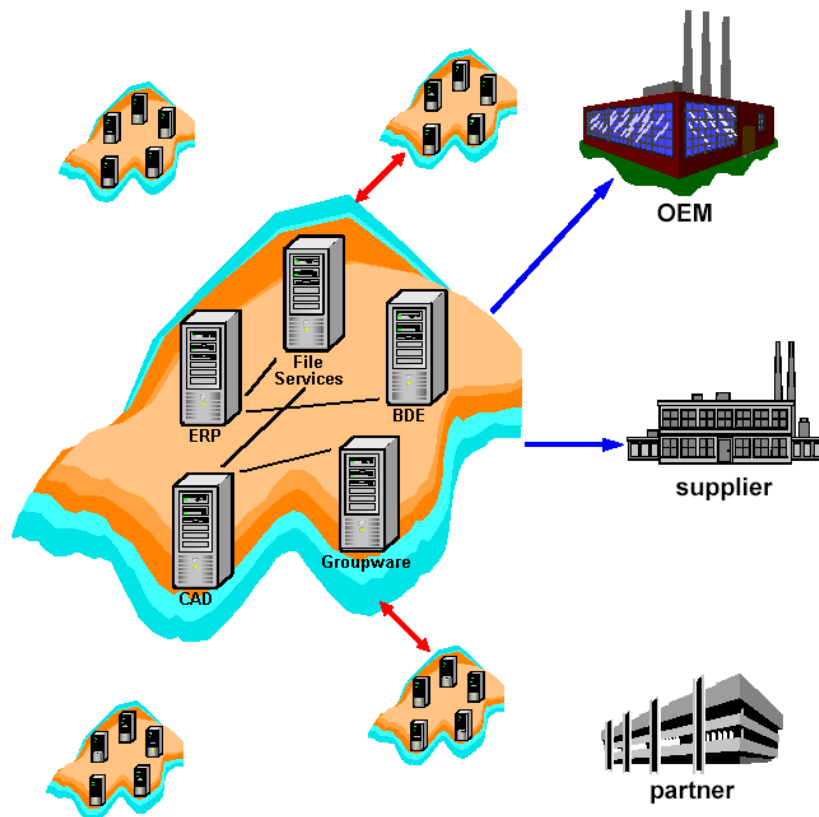


Abbildung 2.11: IT-Systemlandschaft mit nur teilweise verbundenen Anwendungen

Schematisch ergibt sich dadurch ein in der Abbildung 2.11 dargestellter Zustand zwischen unternehmenseigenen Anwendungen wie ERP-System, BDE, CAD usw. und daran angrenzenden Systemen (OEMs, Zulieferer und andere Partner bzw. entfernte Unternehmenseinheiten).

2.8 Fazit

In den vorangegangenen Kapiteln wurden die unterschiedlichen Rahmenbedingungen erläutert, welche die heutige Situation in der Automobilindustrie prägen. In der nachstehenden Tabelle 1 sind die beschriebenen Probleme den praktischen Auswirkungen nochmals gegenübergestellt.

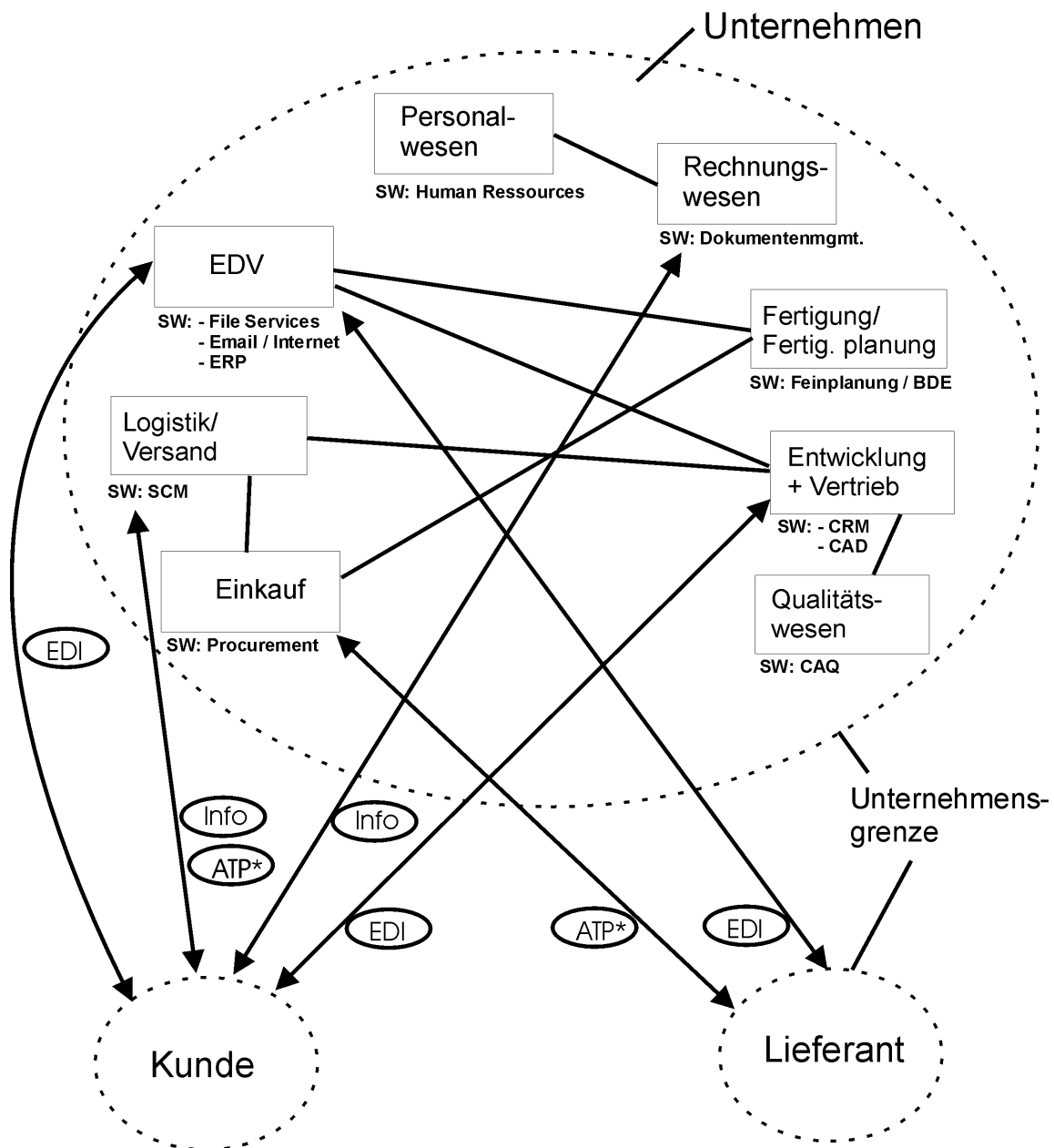
Problem	Auswirkung
Werkszentrische Sichtweise auf Unternehmensprozesse	Keine Berücksichtigung der Auswirkungen auf Supply Chain als Gesamtprozess
Funktionale Unternehmensorganisation	Schnittstellenprobleme zwischen Mitarbeitern und Organisationseinheiten
Globalisierungszwang	Veränderung von Produktion, Modellauswahl, Distribution, Internationalisierung, Preisdruck, Verdrängungswettbewerb
Steigende Nachfrage bei Nischenprodukten	Notwendigkeit einer flexiblen Produktion
Veränderungen der Vertriebsstrukturen	Wandel des Automobils zum Konsumgut
Weltweite Überkapazitäten	Zunehmender globaler Wettbewerb
Verkürzung der Produktentwicklungs- und -lebenszyklen	Innovationsdruck auf Produkt- und Prozessebene
Verkürzung der Produktionszeiten („14-Tage-Auto“)	Veränderung der unternehmensinternen und externen logistischen Prozesse
Konsolidierung	Veränderung der Markt- und Wettbewerbsbedingungen, Kooperationszwang
Abnehmende Fertigungstiefe bei den OEM's	Verschiebung von Wertschöpfungsanteil und Produktrisiko zu den Zulieferern
Modellbezogene Plattformkonzepte in der Produktion, Modulbauweise	Sinkende Auftragszahlen, steigende Auftragsvolumina
Wandel vom Anbieter- zum Käufermarkt (from push to pull)	Veränderungen bei kundenseitigem Preis- und Qualitätsbewusstsein sowie Einkaufsverhalten, „Losgröße 1“, Individualisierung
Lager- und Sicherheitsbestände	Kapitalbindung
Kundenseitig isolierte Bedarfsverläufe	Bedarfsaufschaukelung („Bull-Whip-Effekt“), Sicherheitsbestände
Nicht standardisierte Datenaustausch- und Kommunikationsstrukturen inner- und außerhalb der Zulieferuntern. und OEM's, fehlende Schnittstellen	Zeitverluste, Unterbrechung des Informations- und Datenflusses, Datenredundanz, fehlerhafte Datensätze durch unzureichende Konvertierung
Proprietäre Systeme und Kommunikationsstrukturen	Hoher administrativer Aufwand, hohe Implementierungskosten, Fehleranfälligkeit
Nicht integrierte Systemumgebungen zur Steuerung der Unternehmensprozesse	Nur eingeschränkte Kommunikation zwischen den Prozesssystemen möglich

Tabelle 1: Zusammenfassung des Ist-Zustandes

Zusammenfassend ist festzustellen, dass sich in der Automobilindustrie die überwiegend mittelständischen Zulieferer und die Hersteller aufgrund der Sättigung der Märkte und der Globalisierung in einem Spannungsfeld aus Kundenanforderungen, Wettbewerb, Gewinnerzielung und Leistungsfähigkeit befinden. Durch eine stetige Verkürzung der Produktlebenszyklen und einer größeren Variantenvielfalt bei kleineren Losgrößen tritt die Flexibilität bei den Fertigungsverfahren immer stärker in den Vordergrund, da starre Produktionsstrukturen den Anforderungen nicht mehr gerecht werden können. Gleichzeitig bieten die Unternehmensorganisation und die logistischen Unternehmensprozesse ein großes Optimierungspotential, wozu ein Wechsel zur prozessorientierten und werksübergreifenden Sichtweise notwendig ist. Hierbei liegt ein Schlüssel in der Gestaltung der unternehmensinternen und -externen Informations- und Kommunikationsstrukturen, die bislang von Heterogenität und Inkompatibilität geprägt sind: zahlreiche isolierte Softwareanwendungen bilden innerhalb einzelner Unternehmensbereiche separate Kommunikationsinseln und auch die damit abgebildete Prozessstruktur ist aufgrund der funktionalen, nicht prozessorientierten Ausrichtung uneinheitlich und schwierig zu beherrschen.

An den Unternehmensgrenzen tritt zu externen Partnern wie Kunden und Lieferanten die gleiche Problematik auf. Hier findet über unterschiedlichste Kommunikationskanäle (EDI, telefonische Informationen, Email usw.) die Abbildung einzelner isolierter Teilprozesse statt (z. B. Verfügbarkeitsabfragen (**ATP**⁵⁰) oder Anfragen zu Produktinformationen). Diese Situation ist zusammenfassend in der Abbildung 2.12 dargestellt.

⁵⁰ Available To Promise



ATP* = ATP-Anfrage, z. B. telefonisch

Abbildung 2.12: heutige Situation im mittelständischen Automotive-Bereich

3 Anforderungen für eine Anwendungsintegration

3.1 Anforderungen aus der Sicht der Unternehmen

Die Analyse des Ist-Zustandes hat gezeigt, dass für die Probleme in der mittelständischen Automobilindustrie aufgrund der Komplexität, die sich über Unternehmensorganisation, Unternehmensstrategie und Technologieveränderungen bis hin zu nationalen und internationalen Marktveränderungen erstrecken, keine pauschale Lösung existieren kann. Für mögliche Lösungsszenarien lassen sich jedoch eine Reihe von Anforderungen ableiten, die auf Grund der Zielsetzung dieser Arbeit aus der Sicht der Informationstechnologien beschrieben werden. Dies ist umso naheliegender, als sämtliche Unternehmensaufgaben und –prozesse sowohl intern als auch extern zunehmend durch IT-Systeme unterstützt bzw. abgebildet werden. Die Grundannahme ist dabei, dass den äußeren Zwängen wie Preisdruck und Globalisierung, die von den Märkten vorgegeben werden und seitens der einzelnen Unternehmen kaum beeinflussbar sind, vor dem Hintergrund einer nachhaltigen Wettbewerbsfähigkeit nur mit einer effizienteren Prozessgestaltung entgegengewirkt werden kann. An dieser Stelle ist im Rahmen von Prozessoptimierungen ein erhebliches Rationalisierungspotential zu erreichen durch den Wandel von funktionsorientierten hin zu prozessorientierten Abläufen. Hierbei können die Verschwendung in Form von Überproduktion, eine unnötige Lagerhaltung, auftretende Wartezeiten, Ausschuss, zusätzliche Materialbewegungen und Transporte sowie überflüssige Bearbeitungsschritte erheblich reduziert werden [96].

Bei der Betrachtung der Unternehmensprozesse an den Schnittstellen zu internen und externen Strukturen wird deutlich, dass sich im Rahmen von Prozessoptimierungen über Optimierungen von Teilprozessen nicht automatisch die Optimierung des Gesamtprozesses erreichen lässt [83]. Für die internen Prozesse bedeutet dies, dass sie parallel zu den externen Prozessen, aber unter Berücksichtigung der gemeinsamen Zielsetzung betrachtet werden müssen, was gleichzeitig ein Grundprinzip des Supply Chain Managements darstellt. Im weiteren Sinne wird unter Supply Chain Management die integrierte prozessorientierte Planung und Steuerung der Waren-, Informations- und Geldflüsse entlang der gesamten Wertschöpfungskette vom Rohstofflieferanten bis zum Kunden verstanden, und zwar mit den Zielsetzungen einer verbesserten Kundenorientierung, einer Synchronisation der Versorgung mit dem Bedarf, einer flexiblen und bedarfsgerechten Produktion sowie einem Abbau der Bestände entlang der Wertschöpfungskette. Diese Ziele stellen somit effiziente Lösungsansätze für die beschriebenen Probleme der Automobilindustrie aus Kapitel 2 dar

und sind mit geeigneten Systemen umzusetzen. Um dabei der für die mittelständische Industrie erforderlichen Kostenstruktur gerecht zu werden, ist die kostengünstige Integration in die bestehenden Systeme sowie ein geringer Implementierungs- und Administrationsaufwand erforderlich. Hierzu sind skalierbare modulare Ansätze geeignet, die die schrittweise wachsenden Integrationsanforderungen berücksichtigen, indem sich die Integrationspfade reibungslos in den Prozessablauf einbinden lassen. In diesem Zusammenhang muss auch im Rahmen des Geschäftsprozess-Managements die Anpassung der Integrationsanforderungen an die Unternehmensprozesse möglich sein, mit der Maßgabe, so viel Integrationsaufwand wie nötig, jedoch so wenig wie möglich zu betreiben. Dadurch wird die Anschaffung teurer Komplett-Lösungen, wie sie in Großunternehmen benötigt werden, vermieden.

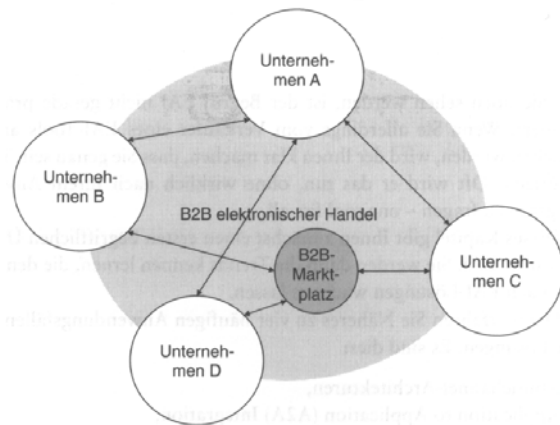
3.2 Anforderungen an die Kommunikation

Die grundlegenden Bestandteile der heutigen Unternehmensprozesse sind IT-Lösungen, die mehr oder weniger stark miteinander verzahnt sind. Die Kommunikation zwischen diesen Systemen bzw. deren Integration spielt im Rahmen der SCM-Betrachtung jedoch eine zentrale Rolle [97], wobei zunächst die interne von der externen Integration zu unterscheiden ist. Mit interner Integration ist die Integration zwischen den Kernanwendungen (Core-Applications) eines Unternehmens gemeint, was auch als **A2A**⁵¹ bezeichnet wird, externe Integration stellt die Basis für die Kommunikation mit externen Partnern wie Kunden und Lieferanten dar (**B2B**⁵²) [98]. In Abbildung 3.1 und Abbildung 3.2 sind schematisch die Zusammenhänge zwischen interner und externer Integration beschrieben, wobei einerseits von der Kommunikation zwischen Anwendungen bzw. Schnittstellen unterschiedlicher Unternehmen und andererseits von der Kommunikation zwischen unterschiedlichen Anwendungen eines Unternehmens ausgegangen wird [99].

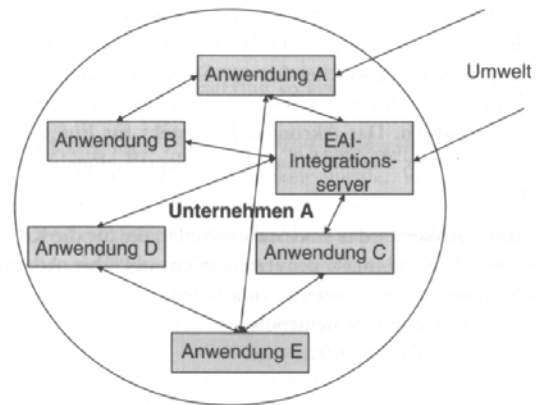
Auch innerhalb des SCM wird unterschieden zwischen internen und externen Bereichen (interne Supply Chain und externe Supply Chain). Beiden gemeinsam sind die Prozesse der Beschaffung, der Herstellung und der Lieferung zu internen oder externen Lieferanten bzw. Kunden.

⁵¹ Application To Application

⁵² Business To Business



**Abbildung 3.1: externe Integration
zwischen Unternehmen [99]**



**Abbildung 3.2: interne Integration
zwischen Anwendungen [99]**

Aus dem Zusammenspiel beider Supply Chains entsteht schließlich die integrierte Supply Chain, über die sämtliche Mittel- und Informationsflüsse (Material, Finanzen), die beteiligten Partner (Kunden, Lieferanten, Dienstleister, Produktion, Versand) sowie die erforderliche Infrastruktur (Prozesse, Werkzeuge, Organisation, Zielsetzung) beschrieben werden. Die Zusammenhänge sind in der Abbildung 3.3 dargestellt [3].

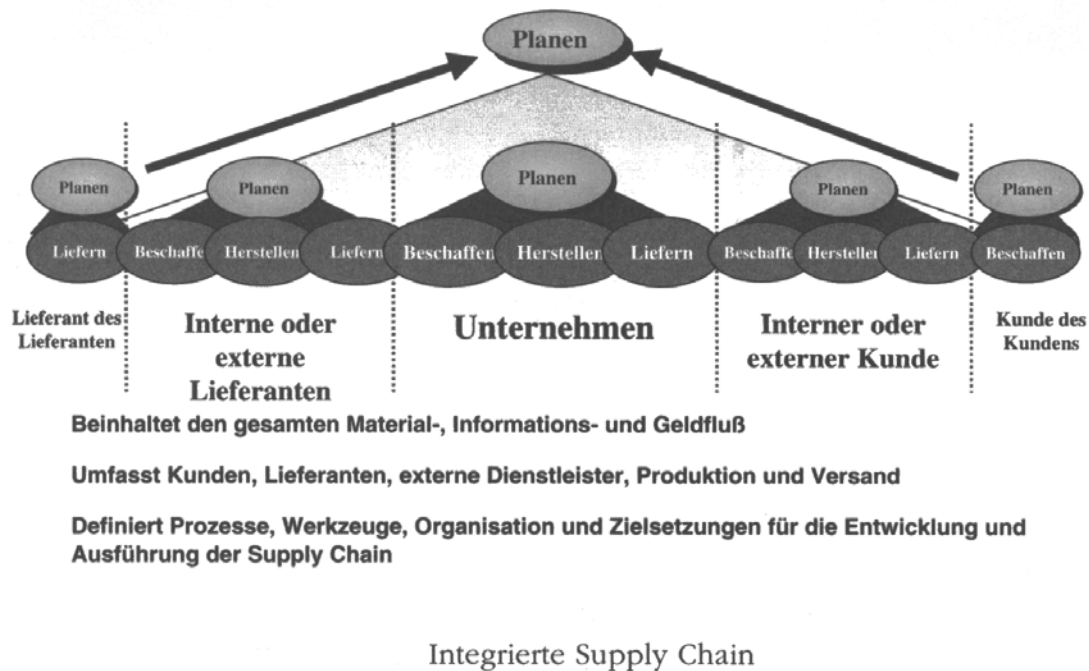


Abbildung 3.3: interne, externe und integrierte Supply Chain [3]

3.3 Anforderungen an die Schnittstellen

Solange an den Schnittstellen von interner und externer Supply Chain der Kommunikationsbedarf des Supply Chain Managements durch interne und externe Integration in den beteiligten Teilsystemen nicht gedeckt wird, sind die Ziele des SCM nur schwer erreichbar. Die Integrationslösungen innerhalb der Informationstechnologien werden daher als Enabler für Supply Chain Prozesse identifiziert. Die aus der Sicht der IT notwendige unternehmensweite Anwendungsintegration stellt somit eine grundlegende Voraussetzung für die integrierte Supply Chain dar. Mit der Anwendungsintegration werden, auch über die eigenen Unternehmensgrenzen hinweg, aus isolierten IT-Anwendungen integrierte Geschäftsprozesse.

Systeme, die diese integrierten Prozesse unterstützen, lassen sich nach ihrem Integrationsgrad einordnen, wobei der Integrationsgrad den Umfang darstellt, in dem Applikationen integriert werden [99]. In der Abbildung 3.4 ist schematisch der Integrationsgrad in Abhängigkeit von der Flexibilität der Integrationssysteme dargestellt.

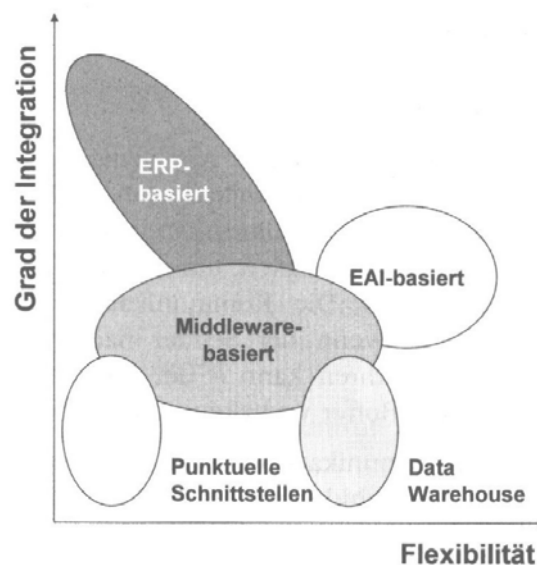


Abbildung 3.4: Integrationsgrad von EAI und traditioneller Integrationsansätze [99]

Demzufolge bieten traditionelle Ansätze wie isolierte Schnittstellen eine geringe Flexibilität bei ebenfalls geringer Integration. ERP-basierte Ansätze lassen häufig keine flexiblen Integrationswege zu, weisen allerdings teilweise hohe Integrationsgrade auf. Demgegenüber stehen Data Warehouses, die aufgrund der Integration auf Datenebene verschiedene Datenbestände verwalten und flexibel sind, jedoch nur in geringem Umfang Integrationspfade abbilden können. Middleware-basierte Ansätze dagegen bieten flexible

integrative Lösungen auf hohem Niveau, sie werden lediglich von EAI-Systemen übertroffen. Mit EAI-Systemen werden die Kosten und insbesondere der Zeitaufwand für die Verbindung von Applikationen, wie es sonst bei proprietären Schnittstellenprogrammierungen der Fall ist, sowie die Komplexität von Integrationsaufgaben reduziert. Da EAI-Lösungen mit zentralisierten Schnittstellensystemen bei der Anwendungsintegration funktionell die beste Lösung darstellen, sollen deren Eigenschaften im folgenden beschrieben werden.

3.4 Anforderungen im Zusammenhang mit der Integration

Auf dem Weg zur Umsetzung von SCM-Konzepten ist die Integration der IT-Anwendungen, die Daten an die beteiligten Partner innerhalb der internen und externen Supply Chain weitergeben müssen, Voraussetzung. Die Abbildung 3.5 illustriert beispielhaft ein mögliches Integrationsszenario einer internen und externen Integration zwischen Lieferant und Kunde [100].

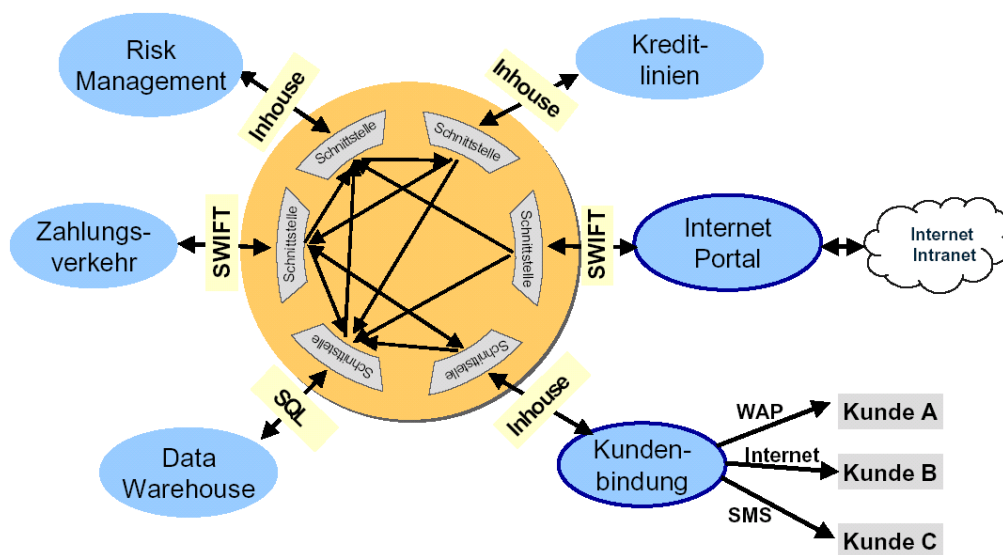


Abbildung 3.5: Integrationsszenario für interne und externe Integration [100]

Dargestellt sind hier typische Unternehmensanwendungen wie Datenbanken und Internetportal, Anwendungen für Zahlungsverkehr und Kreditlinien, Risk Management und

Anwendungssysteme für Kundenbindungen mit einem zentralen EAI-System, über das die Kommunikation auf unterschiedliche Weise erfolgen kann (**SWIFT**⁵³, **SQL**⁵⁴, **WAP**⁵⁵, Internet, **SMS**⁵⁶ oder spezielle Inhouse-Formate). Oftmals werden diese Systeme auch als „Informationsdrehscheibe“ bezeichnet. Die in diesem Zusammenhang erforderlichen generellen Eigenschaften von EAI-Systemen sind [99, 101]

- Transport von Nachrichten: EAI-Infrastrukturen müssen in der Lage sein, Nachrichten in Echtzeit zu transportieren, damit diese den beteiligten SCM-Partnern direkt ohne Verzögerung vorliegen. Hierbei sind **P2P**⁵⁷-Infrastrukturen denkbar, die nur wenige zentrale Services anbieten und bei denen die eigentlichen Messages nicht über einen zentralen Server laufen (Bsp. **DCE**⁵⁸) sowie Bus-Infrastrukturen wie **CORBA**⁵⁹, bei denen alle Nachrichten in interne Normalformen konvertiert werden, und EAI-Server, bei denen die Nachrichten über zentrale Server laufen und die mit einer Hub & Spoke⁶⁰-Architektur vergleichbar sind.
- Unterschiedliche Kommunikationsstile: Kommunikationsstile lassen sich in synchrone und asynchrone Übertragung unterscheiden. Bei synchroner Kommunikation sind Sender und Empfänger in ihrem Ablauf aneinander gekoppelt. Asynchrone Kommunikation ist gegeben, wenn der Empfänger nicht bekannt ist, wie dies beispielsweise bei Broadcasting der Fall ist.
- Genormte Nachrichtenformate: Neben genormten Basisformaten, die **RPC**⁶¹-basierte Systeme wie DCE und CORBA einsetzen, ist **XML**⁶² das bedeutendste Format. XML kann genutzt werden als Format für den Inhalt von Nachrichten, zur Vereinheitlichung der Kommunikation im Unternehmen sowie von unternehmensübergreifenden Strukturen, wobei über eine Normungsinstanz in Form von Schemata das Format der Metadaten vorgegeben werden muss.

⁵³ Society for Worldwide Interbank Financial Telecommunications

⁵⁴ Structured Query Language

⁵⁵ Wireless Application Protocol

⁵⁶ Short Message Service

⁵⁷ Punkt-zu-Punkt

⁵⁸ Distributed Computing Environment

⁵⁹ Common Object Request Broker Architecture

⁶⁰ Nachrichten werden von einer zentralen Stelle (Hub) empfangen, transformiert und weitergeleitet

⁶¹ Remote Procedure Call

⁶² eXtensible Markup Language

- **Datentransformation:** In Unternehmen sind häufig Altsysteme im Einsatz, zu denen keine Schnittstellen oder Möglichkeiten zur Datenkonvertierung existieren. Hier müssen entweder individuell erstellte Programme oder ETL-Tools eingesetzt werden, die das Extrahieren von Daten aus Schnittstellen und Datenbanken, das Transformieren der Daten und das Laden in eine Data-Warehouse-Datenbank erleichtern bzw. unterstützen.
- **Unterstützung von Geschäftslogik, -prozessen und -regeln:** Mitunter ist es erforderlich, dass Daten oder Services aus verschiedenen Systemen durch zusätzliche Funktionalitäten ergänzt werden, um auf diese Weise die geforderte Geschäftslogik abzubilden. Diese Funktionalitäten sind durch geeignete Werkzeuge bereitzustellen, typischerweise durch eine flexible Programmier- und Beschreibungsumgebung. Dabei spielt die Umsetzbarkeit der Gesamtprozesse, wie sie beispielsweise durch Workflows beschrieben werden, eine wichtige Rolle.
- **Konnektivität:** Die Einbindung möglichst aller vorhandenen Systemumgebungen über entsprechende Adapter in die EAI-Struktur ist eine wesentliche Voraussetzung (z. B. Daten- und Dateistrukturen). Zu berücksichtigen sind hier auch externe Systeme bei Kunden oder Lieferanten. Nur so lassen sich Heterogenität und Systemvielfalt transparent gestalten.
- **Effiziente Ressourcennutzung:** Idealerweise setzen EAI-Systeme auf vorhandene Systeme auf, so dass vorhandene Hard- oder Software-Plattformen genutzt werden können. Diese Vorgehensweise wirkt einer Vergrößerung der Heterogenität durch weitere Systeme entgegen.
- **Sicherheitskonzepte:** Bei der Kommunikation zwischen den beteiligten Systemen müssen Verfahren zur Authentifizierung und zur Verschlüsselung der Informationen berücksichtigt werden können.
- **Scheduling von Nachrichten und Routing:** Informationen müssen über definierte Wege an die betreffenden Empfänger weitergeleitet werden können.
- **Load Balancing, Failover, Monitoring, Recovery-Fähigkeit, Tracing und Debugging:** Diese Eigenschaften sind nicht unbedingt spezifisch für EAI-Systeme, sondern grundsätzliche Anforderungen an zeit- und geschäftskritische Systeme. Dazu zählen

ebenfalls Merkmale wie Protokollierung, Performance, Skalierbarkeit und Zuverlässigkeit der Anwendung.

Im Rahmen einer EAI-Referenzstruktur lassen sich die erforderlichen Eigenschaften vereinfacht in drei Schichten einteilen: in eine Prozessschicht, eine Kommunikations- und eine Protokoll-Adapterschicht [101] (vgl. Abbildung 3.6).

In der Prozessschicht werden komplexe Abläufe aus einfachen Kommandos und Aufrufen zusammengesetzt. Hier kann auch der Nachrichtenversand und –empfang zwischen Anwendungen oder Anwendungssystemen initiiert werden. Die Kommunikationsschicht bildet den Nachrichtentransport zwischen Sender und Empfänger ab und ist verantwortlich für das notwendige Routing. Sie ist ebenfalls verantwortlich für die Sicherheit von Aufrufen und die Konvertierung von Nachrichten in verschiedene Formate.

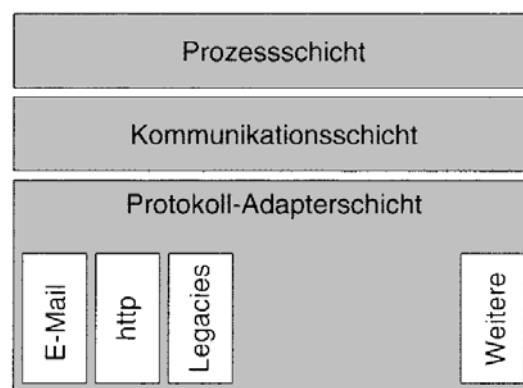


Abbildung 3.6: EAI-Referenzstruktur [101]

Die Protokoll-Adapterschicht schließlich hat die Aufgabe, die über externe Auslöser wie Email, **http**⁶³-Requests, Funktionsaufrufe und Datenbankupdates initiierten Vorgänge in Nachrichten umzusetzen. Ein Protokolladapter kann dabei auch ein fremdes System sein, dem Nachrichten zugestellt werden sollen.

Während die EAI-Referenzstruktur den grundsätzlichen strukturellen Aufbau von EAI-Systemen beschreibt, ist damit noch nichts darüber gesagt, wie die Anwendungsintegration technologisch erfolgt. Unter Zugrundelegung der Schichtenarchitektur für betriebliche Informationssysteme, wie sie von Denert [102] beschrieben wird, ergeben sich dabei nachfolgende Integrationsmethoden (Abbildung 3.7):

⁶³ hypertext transfer protocol

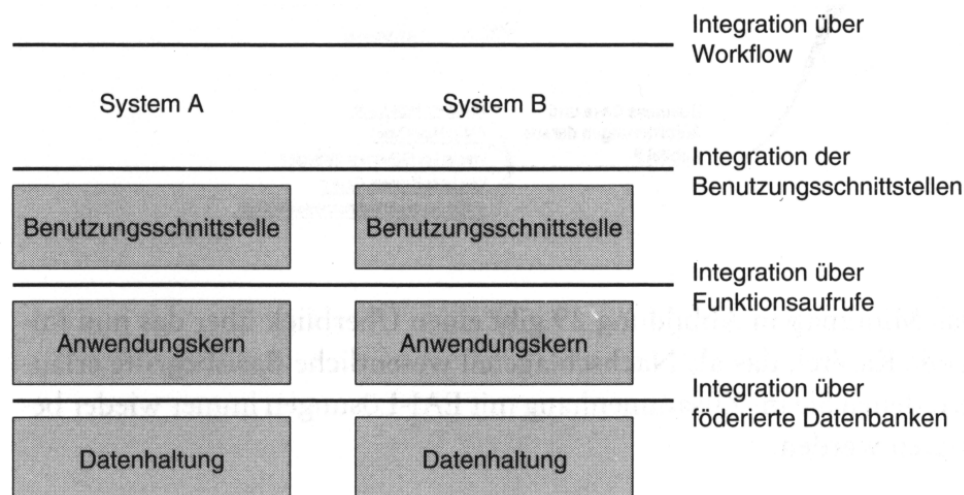


Abbildung 3.7: Integrationsmethoden für die Anwendungsintegration [101]

- **Integration über Benutzerschnittstelle:** Hierbei werden auf die Benutzerschnittstellen existierender Anwendungen neue Benutzerschnittstellen aufgesetzt, mit der Zielsetzung, eine bessere Nutzbarkeit oder erweiterte Funktionalitäten zur Verfügung zu stellen, die sich aus den darunterliegenden Benutzeroberflächen ergeben. Webbasierte Technologien stellen einen vielversprechenden Ansatz dar, sie werden beispielsweise über Frames oder Portalserver realisiert, über die personalisierte Inhalte aus mehreren Systemen zusammengestellt werden. Eine andere Möglichkeit ist das Screen Scraping, bei dem ein Terminal-Datenstrom für andere Zwecke als die Anzeige auf einem Hostterminal aufbereitet wird, etwa in Form einer **GUI**⁶⁴, oder die Integration über Workflow-Manager.
- **Integration über Funktionsaufrufe:** Wesentliches Merkmal der Integration über Funktionsaufrufe ist der Aufruf eines Anwendungskerns über einen anderen Anwendungskern. Dies geschieht typischerweise über Schnittstellen, wobei die Anwendungen auf einem System oder auf verteilten Systemen laufen können.
- **Integration über Datenbanken und Dateistrukturen:** Bei der Integration über Datenbanken kommunizieren mehrere Anwendungen über eine gemeinsame Datenbasis. Dabei ist der Einsatz von föderierten Datenbanken denkbar, bei denen eine Anwendung mehrere Datenbanken benutzt, die über Middleware zu einer

⁶⁴ Graphical User Interface

einigen Datenbank zusammengefasst sind, oder die Nutzung einer gemeinsamen Daten- bzw. Dateistruktur, die von mehreren Anwendungen verwendet wird.

- Integration über Komponenten: Komponenten stellen Programme mit definierten Schnittstellen dar. Sie können in Form von Plug-Ins, Skriptsprachen oder Containern realisiert werden.

EAI-Systeme, welche die beschriebenen Eigenschaften besitzen und Methoden unterstützen, sind als Basis für systemübergreifende SCM-Konzepte zu sehen. Die interne Supply Chain wird dadurch ebenso unterstützt wie die externe, wobei die Daten sowohl innerhalb der Unternehmensgrenzen verwendet als auch Kunden und Lieferanten an die internen Systeme angebunden werden. Welche der Eigenschaften im Einzelfall benötigt werden, hängt maßgeblich von den Integrationsanforderungen ab. Hier spielen neben den technologischen Aspekten auch die durch die zu integrierenden Prozesse vorgegebenen Rahmenbedingungen eine wesentliche Rolle.

3.5 Zusammenfassung der Anforderungen

Die heutige Situation der Hersteller und Zulieferer in der Automobilindustrie ist durch Eigenschaften wie Globalisierung und Preisdruck geprägt. Diese Entwicklungen werden jedoch vom Markt vorgegeben und sind seitens der Unternehmen kaum zu beeinflussen, eine pauschale Lösung der Probleme existiert daher nicht. Diejenigen Stellgrößen, mit denen den Problemen planbar entgegengewirkt werden kann, konzentrieren sich somit auf die Unternehmen selbst, und zwar in organisatorischen und informationstechnischen Lösungsansätzen.

Neben dem Weg zur prozessorientierten Unternehmensorganisation ist die Unterstützung der organisatorischen Maßnahmen durch eine Beschleunigung der unternehmensinternen und –externen Prozesse erforderlich. Prozessorientierung einerseits und Prozessbeschleunigung andererseits werden durch die Verbindung aller beteiligten Prozesspartner erreicht, was letztlich die Steuerung der gesamten Prozesskette ermöglicht. Innerhalb der Wertschöpfungskette werden diese Steuerungsgrößen durch das Supply Chain Management beschrieben. Da bereits heute die meisten Aufgaben und Prozesse rechnergestützt gesteuert werden, ist für ein funktionierendes Supply Chain Management die Kopplung der IT-Systeme Voraussetzung, welche die benötigten Daten an die beteiligten Supply Chain Partner intern und extern übertragen. Diese Daten stammen oftmals aus unterschiedlichen

Teilsystemen, die miteinander nicht verbunden sind, weshalb ohne deren Integration keine Kommunikation möglich ist.

Hier setzen EAI-Systeme an, die aus der Koexistenz einer Vielzahl von unterschiedlichen IT-Lösungen die benötigten Daten den Prozesspartnern zur Verfügung stellen können. Kerngedanke ist dabei die Integration der Unternehmensanwendungen, so dass sich die IT-Systemlandschaft als transparentes homogenes System nach außen hin abbilden lässt. Bei der Definition der Anforderungen lassen sich grundlegende Eigenschaften der benötigten EAI-Systeme von den möglichen Integrationspfaden unterscheiden. Neben der Grundfunktion, Nachrichten transportieren zu können, müssen EAI-Systeme synchrone und asynchrone Kommunikationsstrukturen unterstützen. Die Notwendigkeit der Transformationen von Datenformaten beim Datenaustausch unter Verwendung von genormten Nachrichtenformaten führt zur Forderung der Unterstützung von Geschäftslogiken und -prozessen, die teilweise mit erweiterten Funktionalitäten abzubilden sind.

Schließlich ist die Konnektivität zu möglichst vielen vorhandenen Systemen eine wichtige Integrationsvoraussetzung, ebenso wie die Realisierung von Sicherheitskonzepten und den sonst üblichen Softwareeigenschaften für geschäftskritische Transaktionssysteme wie Load Balancing, Failover, Monitoring, Recovery-Fähigkeit, Tracing und Debugging. Besonders zu berücksichtigen ist im Hinblick auf die Zielsetzung dieser Arbeit die Eignung für den automotiven Mittelstand, was zu den Anforderungen einer flexiblen und kostengünstigen Implementier- und Administrierbarkeit der eingesetzten Lösung führt. In diesem Zusammenhang ist die Implementierung unter Verwendung bereits bestehender IT-Systemen vorteilhaft. Schließlich steht im Rahmen dieser Betrachtung die Abbildung der für das Supply Chain Management erforderlichen Konzepte im Vordergrund, so dass die Integrationsmechanismen die Ziele des SCM verfolgen müssen.

Bei den Integrationsmethoden lassen sich vier Fälle unterscheiden, wobei von der Integration über eine Benutzerschnittstelle der Weg über Funktionsaufrufe ebenso möglich ist, wie die Integration über Datenbanken oder Komponenten. Im allgemeinen wird dabei nicht eine einzige Variante existieren, sondern eine Kombination dieser Möglichkeiten.

4 Grundlagen der eingesetzten Informationstechnik

Zur Umsetzung der beschriebenen Lösungsanforderungen aus dem vorherigen Kapitel sind nun die Grundlagen der erforderlichen Informationstechnik zu beschreiben. Hierbei stellt sich die Frage, ob sich die gesamten Forderungen mit einem bereits auf dem Markt verfügbaren Software-System lösen lassen oder ob es einer individuellen Programmierung in Form einer Spezialsoftware bedarf. Der zweite Lösungsweg würde zwar die größte Übereinstimmung mit den Forderungen bringen, ist aber wegen der hohen Kosten und der sicherlich langen Entwicklungszeit uninteressant. Falls keines der bereits vorhandenen EAI-Systeme geeignet ist, verbleibt insofern nur ein so genannter Mittelweg. Das heißt, dass die Lösung dann lauten muss: Nutzung und Kombination vorhandener Standard-Komponenten mit Hilfe einer geeigneten Programmierumgebung.

Zunächst werden die existierenden EAI-Systeme im Überblick betrachtet und den jeweiligen Architekturtypen zugeordnet. Eine umfassende Auseinandersetzung mit den EAI-Systemen folgt im Konzept in Kapitel 5, wobei auch auf deren Probleme sowie deren Eignung für die mittelständische Industrie eingegangen wird. Anschließend an die Übersicht der EAI-Systeme werden die möglichen Programmierumgebungen für einen Lösungsweg aufgezeigt. Dies ist notwendig, da kein vorab entwickeltes Standard-Softwaresystem so konzipiert ist, dass dieses ohne jegliche Anpassung an die vorhandenen Prozesse in einem Unternehmen eingesetzt werden kann.

4.1 Verfügbare EAI-Systeme

Der Markt für EAI-Produkte ist von einer großen Dynamik geprägt, in dem ebenfalls ein Trend zur Konsolidierung erkennbar ist [99]. Zur Übersicht über die vorhandene Vielfalt ist in der Abbildung 4.1 die Aufteilung des nordamerikanischen EAI-Marktes im Jahr 2001 dargestellt.

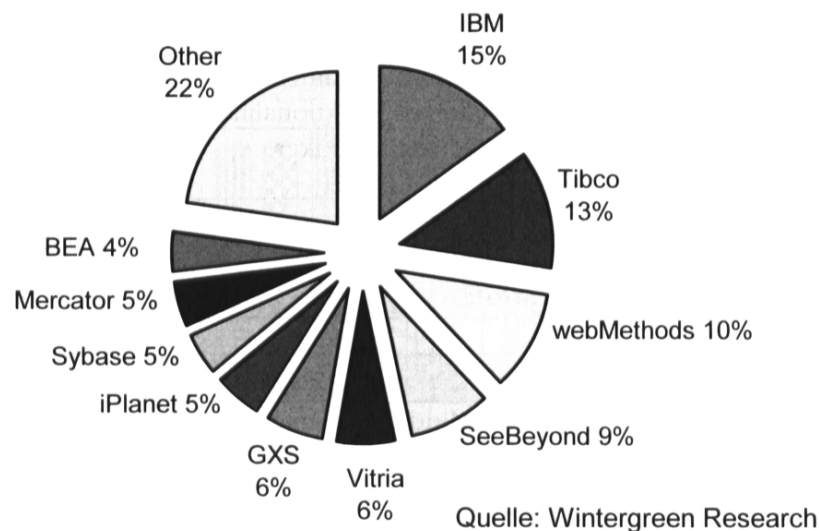


Abbildung 4.1: Marktanteile EAI-Anbieter, USA 2001 [99]

Eine Einteilung der Systeme erfolgt in der Literatur nach drei Arten, nämlich nach dem Ebenenmodell, der Integrationsmethode und der EAI-Architektur [100]. In Anlehnung an Kapitel 3 wird beim Ebenenmodell in Transportebene, Konvertierungsebene, Prozessebene und externe Integrationsebene unterschieden. Technologisch interessanter sind jedoch die beiden EAI-Systemklassifizierungen nach Integrationsmethode und EAI-Architektur, die nachfolgend näher beschrieben werden.

Die Einteilung nach der Integrationsmethode unterteilt die Systeme in solche zur Integration auf Datenebene, auf API-Ebene, Prozessebene und Userebene. Unter Integration auf Datenebene (Data Level EAI) werden der Prozess und die Technologie zum Datenaustausch zwischen Datenbanken verstanden. Beliebige Datenbanken können Daten miteinander austauschen und nutzen dafür Middleware oder das Internet als Transportsystem. Hierbei dienen Datenintegrations-Tools als intelligente EAI-Software. Integration auf API-Ebene bezieht sich auf die Anbindung von Inhouse- bzw. Standardanwendungen auf Basis vorhandener Schnittstellen (API's) ohne manuelle Eingriffe. Diese API's können je nach Spezifikation des Herstellers u. a. sowohl für den Zugriff auf Daten als auch auf Geschäftsprozesse genutzt werden. API's können im einfachsten Fall aber auch Dateien (Flatfiles) einbinden, die in einer Verzeichnisstruktur geschrieben oder gelesen werden.

Integration auf Prozessebene (Process Level EAI) bedeutet die gemeinsame Nutzung von Business-Logik. Beispielsweise kann die Methode zur Aktualisierung eines Kundendatensatzes von unterschiedlichen Applikationen gemeinsam genutzt werden, wofür

Methoden wie Distributed Objects, Application Servers und **TP**⁶⁵ Monitore verwendet werden können. Unter Integration auf Anwenderebene (Userinterface Level EAI) wird der Zugriff auf Daten aus Usermasken verstanden („Screen Scraping“). Dies kann z. B. bei Mainframe-Anwendungen auftreten, die keinen Zugriff über Datenebene, API-Ebene oder Prozessebene ermöglichen. Die Einteilung nach Methode bzw. Typ der Integration deckt sich dabei auch mit dem Ebenenmodell, da bei den verschiedenen Methoden immer die drei Ebenen Transport, Konvertierung und Prozesse betrachtet werden müssen und auf der anderen Seite alle diese Methoden in einer EAI-Lösung zur Integration der verschiedenen Anwendungen vorkommen können.

4.2 Klassifizierung von EAI-Systemen

Bei der Klassifizierung nach EAI-Architekturen werden die grundsätzlichen Funktionalitäten durch vier Architekturtypen beschrieben:

- Processware (Process Integration Server),
- Message Broker,
- Middleware Integration und
- Datenintegration.

Processware ist durch ihre zentrale Ausrichtung als “Server-Centric-Lösung” definiert. Im Mittelpunkt steht ein skalierbarer Integrations-Server als Multitier-Anwendung, der alle Aufgaben eines Application-Servers wahrnimmt und zusätzlich weitere Funktionen bietet wie

- Automatisierung von Workflows,
- regelbasierte Engine,
- Datentransformation,
- Event-Schnittstelle und
- vorgefertigte Adapter.

Aus Sicht von EAI und der Business-Prozesse steht dadurch eine vollständig implementierte Lösung zur Verfügung. Processware kann als Lösung für komplexe und auch einfachere Integrationsanforderungen zwischen einer Reihe von Anwendungen (ERP, Legacy, CRM, SCM, etc.) via Message-Middleware oder Internet und Einsatz aller Standards wie XML oder

⁶⁵ Transaktionsprozess

branchenspezifische Definitionen wie RosettaNet dienen. Sie ist besonders interessant, wenn vorhandene Software-Komponenten mit neuen Software-Komponenten integriert werden können (CORBA, **DCOM**⁶⁶-Objekte).

Message Broker werden als "Network-Centric" beschrieben, weshalb auf allen erforderlichen Plattformen im Netz eine Run-Time-Umgebung installiert ist. Ein Sender schickt eine Nachricht ins Netz und die Run-Time-Umgebungen prüfen, ob die Nachricht für sie bestimmt ist. Um unnötigen Overhead zu vermeiden, werden verschiedene Kommunikationstypen unterschieden: Nachrichten für einzelne und adressierbare Empfänger, Empfängergruppen als Publish/Subscribe-Mechanismus und freie Nachrichten für alle interessierten User. Diese Architektur ist gekennzeichnet durch das Fehlen eines zentralen Integrationservers. Dessen Funktionen mit allen definierten Integrationsfunktionen (Konnektoren, Datentransformation und regelbasierter Engine, Process-Workflow) wird von dezentralen Funktionsbausteinen übernommen, die mehrfach im Netz vorhanden sind. Ein zentrales Directory mit Replikationsfunktionalität übernimmt die Steuerung der zusammenhängenden Abläufe.

Durch Adapter oder Konnektoren bietet auch eher technisch-orientierte Middleware die Integration mit Anwendungen. Im allgemeinen werden hierbei Hub- und Spoke-Architekturen realisiert, die wiederum durch manche Hersteller als Message Broker bezeichnet werden. Durch funktionale Erweiterungen der Middleware-Produkte lassen sich bereits vorhandene Anwendungen (z. B. ERP- und CRM-Systeme) anschließen. Auch moderne EAI-Tools greifen immer wieder auf Middlewareprodukte wie z. B. dem RPC zu. Diese Aufrufe werden allerdings durch TP-Monitore oder Application Server verdeckt.

Datenintegration schließlich ist gekennzeichnet durch Kommunikation zwischen Datenbanken mit Datentransformation zwischen Business Objekten, Polling-Mechanismus oder **DB**⁶⁷-Trigger für eine Event-Steuerung. Es erfolgt die Integration via Datenbanken durch Datentransport und Umformung der Daten entsprechend einem Regelwerk. Damit können redundante DB-Inhalte mittels Events automatisch synchronisiert werden. Auch diese Lösungen basieren in der Regel auf Hub-and Spoke-Architekturen.

Von den zum Zeitpunkt der Untersuchung auf dem Markt angebotenen EAI-Systemen sollen im Rahmen des hier zu entwickelnden Konzeptes 17 Systeme einer näheren Betrachtung unterzogen werden. Da die spätere Realisierung im Zusammenhang mit einem

⁶⁶ Component Object Model

⁶⁷ Datenbank

Unternehmen der Automobilindustrie durchgeführt wird, sind bei der Auswahl für das Konzept diese Kriterien gesondert zu berücksichtigen. Nachfolgend sind in der Tabelle 2 diese 17 Systeme aufgeführt und den zuvor genannten Architekturen zugeordnet.

Architektur	EAI-System	Hersteller
Processware	1. Alliance Manager 2. CrossWorlds 3. Forté Fusion 4. UNIFACE	Extricity CrossWorlds IPlanet Compuware
Message Broker	1. Active Enterprise 2. Business Ware 3. e*Gate / e*Xchange 4. iWave 5. Mercator 6. MINT 7. ROMA 8. WebMethods	TIBCO Vitria See Beyond NEON Systems Mercator Software SunGard Business Integration ICandle WebMethods
Middleware Integration	1. Constellar Hub	Data Mirror
Datenintegration	1. e-Biz Integrator 2. eLink / Tuxedo / WebLogic 3. MQSeries/MQIntegrator/MQWorkflow 4. BizTalkServer 2002	New Era of Networks BEA Systems IBM Microsoft

Tabelle 2: Architekturen der EAI-Systeme

4.3 Programmierungsumgebung für das Konzept

Vor dem Hintergrund der Komplexität der heute am Markt erhältlichen EAI-Lösungen stellt sich die Frage nach deren Eignung für mittelständische Automotive-Unternehmen, die zum einen die Kostenseite berücksichtigen müssen und zum anderen nur einen Teil der angesprochenen Leistungsfähigkeit komplexer Systeme benötigen (vgl. Kapitel 5.3, Beurteilung der EAI-Systeme zwecks Eignung für das Konzept). Für Anpassungen von EAI-Lösungen an die vorhandenen Gegebenheiten eines Unternehmens bieten sich in diesem Zusammenhang verschiedene Programmierungsumgebungen an. Es sind weltweit etwa 1000 Programmiersprachen bekannt [103], die sich grob in Compilersprachen (z. B. Java, C++)

und Skriptsprachen (**Perl**⁶⁸, Cold Fusion, **ASP**⁶⁹) unterscheiden lassen. Während Kompilersprachen für umfangreiche Programmieraufgaben eingesetzt werden und erst über spezielle Compiler teilweise plattformspezifisch in ausführbaren Programmcode gebracht werden müssen, finden Skriptsprachen auch Anwendung für kleinere Teilaufgaben sowie im Bereich der Webservices und sind bereits als Skript lauffähig. Hierzu benötigen die Programmskripte allerdings einen Interpreter oder Parser, der die Skripte zur Laufzeit interpretiert. Die meisten Skriptsprachen sind serverseitig aufgebaut, d. h. der Parser ist innerhalb einer Client-Server-Umgebung auf einem Server installiert und führt dort die Skripte aus.

Skriptsprachen sind im Allgemeinen leichter zu erlernen als Kompilersprachen, und durch den Wegfall des Kompilierens wird die Software-Entwicklung mit Skriptsprachen beschleunigt. In den letzten Jahren hat v. a. die Skriptsprache **PHP**⁷⁰ an Verbreitung und Bedeutung gewonnen, da sie sich speziell auf die Anwendung für Webgestaltung und –programmierung konzentriert und insofern technologisch eine hervorragende Eignung für die Unterstützung der webbasierten Anwendungsentwicklung besitzt. PHP kann gemeinsam mit **HTML**⁷¹-Code in einem Skript kombiniert werden, ist leicht zu erlernen und bietet eine gute Performance. In der aktuellen Version 5 wurden die Möglichkeiten der Objektorientierung im Vergleich zur vorherigen Version deutlich erweitert. Herausragendste Eigenschaft von PHP ist jedoch der umfangreiche Funktionsumfang, insbesondere für Daten- und Datenbankstrukturen, die entweder nativ oder via **ODBC**⁷² angesprochen werden.

Um den Marktanforderungen hinsichtlich Integration und Kollaboration gerecht zu werden, soll aus diesem Grunde bei Nichteignung der analysierten EAI-Systeme ein neues Lösungskonzept untersucht werden, welches einerseits auf PHP basiert und andererseits auf entweder bereits in den meisten Unternehmen vorhandenen oder aber in die Unternehmensumgebungen leicht implementierbare IT-Komponenten aufbaut. Die für dieses Konzept erforderliche Systemumgebung besteht daher aus

- dem OpenSource-Betriebssystem **SuSE**⁷³ **Linux**, das sich durch eine hervorragende Stabilität sowie seinen geringen Preis auszeichnet,
- dem in der Betriebssystemdistribution enthaltenen Webserver **Apache**,

⁶⁸ Page Extension and Reporting Language

⁶⁹ Active Server Pages

⁷⁰ Hypertext Preprocessor

⁷¹ Hypertext Markup Language

⁷² Open Database Connectivity

⁷³ Software und Systementwicklung

- dem in der Betriebssystemdistribution enthaltenen relationalen Datenbanksystem MySQL sowie
- der ebenfalls in der Betriebssystemdistribution enthaltene Parser zur Skriptsprache PHP 4,

und führen zu einer Lösung, die insgesamt als LAMP-Technologie bezeichnet wird. LAMP wird bereits in großem Umfang hauptsächlich als Plattform für die Generierung dynamischer Webseiten eingesetzt und stammt daher ursprünglich aus dem Bereich der Web- und Internettechnologien [104]. Die webbasierte Anwendungsintegration ist daher das Prinzip bei der Verwendung von LAMP als Integrationsplattform. Ihre besondere Eignung für die Abbildung von Integrationsanforderungen, wie sie für Prozesse des Supply Chain Managements erforderlich sind, wird auf den folgenden Seiten dargestellt.

5 Konzept für eine Anwendungsintegration

5.1 Allgemeines

In den vorangegangenen Kapiteln wurden die heute existierenden Probleme der Automobilindustrie geschildert, und es sind tragfähige Lösungsansätze beschrieben worden, diese Probleme zu lösen. Der Lösungsansatz enthält die Notwendigkeit zu strukturellen Veränderungen auf der Geschäftsprozessebene, wobei als bedeutsamste Geschäftsprozesskategorie die Prozesse des Supply Chain Managements identifiziert wurden. Darunter wird die integrierte prozessorientierte Planung und Steuerung der Waren-, Informations- und Geldflüsse entlang der gesamten Wertschöpfungskette verstanden.

Kern eines Lösungskonzepts sind daher die SCM-Geschäftsprozesse, die in der mittelständischen Automobilindustrie durch das Redesign der bisherigen Geschäftsprozesse zu etablieren bzw. zu verbessern sind. Von zentraler Bedeutung sind in diesem Zusammenhang die Integration sowie die Echtzeitkommunikation der beteiligten Anwendungssysteme, mit denen die SCM-Prozesse abgebildet werden, denn nur durch eine verzögerungsfreie Weiterleitung der in der Supply Chain benötigten Informationen sind die von großer Dynamik geprägten SCM-Prozesse zu beherrschen. Um eine Kommunikation in Echtzeit zu erreichen, müssen die für die betreffenden Daten- und Informationsflüsse erforderlichen Infrastrukturen geschaffen werden. Diese Infrastrukturen werden durch die Enterprise Application Integration hergestellt, indem einzelne verteilte IT-Anwendungen eines Unternehmens zu einem verteilten bzw. vernetzten Anwendungssystem integriert werden. EAI stellt insofern eine notwendige Voraussetzung für das Supply Chain Management dar.

5.2 Analyse der vorhandenen EAI-Systeme

Im Folgenden werden die Details von insgesamt 17 namhaften EAI-Softwareprodukten gemäß Tabelle 2 in alphabetischer Reihenfolge dargestellt, wobei die Ergebnisse einer Marktstudie in diesem Bereich aufgeführt sind [100]. Fast alle der genannten Produkte sind auch in der Abbildung 4.1 enthalten. Der Markt an EAI-Lösungen ist jedoch weitaus größer als sich hier darstellen lässt, und wie bereits erwähnt sehr dynamisch, weshalb an dieser Stelle auf die aktuellen Inhalte des EAI-Forums verwiesen wird (www.eaiforum.de).

5.2.1 TIBCO: Active Enterprise

TIBCO deckt als EAI-Hersteller die Transportebene, die Konvertierungsebene und die Prozessebene mit einer Produktpalette (**TIB**⁷⁴/ActiveEnterprise) ab. Der Schwerpunkt der Integration liegt auf der B2B-Integration über das Internet, B2B-Lösungen werden durch Kommunikation der entsprechenden Anwendungen realisiert. TIBCO positioniert dabei die ActiveEnterprise Suite als real-time Tool. TIBCO Finance hat sich komplett auf den Finanzmarkt spezialisiert.

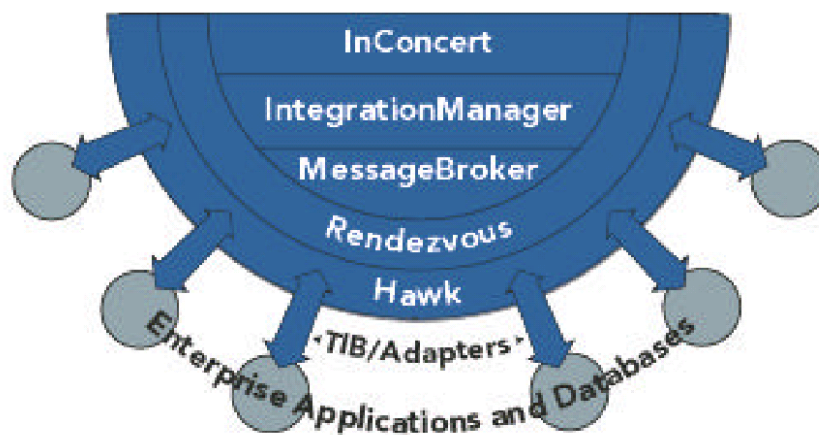


Abbildung 5.1: ActiveEnterprise Architektur [100]

ActiveEnterprise ist auf Basis eines Message Brokers mit konsequenter Umsetzung eines Publish/Subscribe Verfahrens realisiert. Dies bedeutet, die Realisierung eines föderativen Konzeptes ohne zentrale Komponente, was Vorteile bezüglich Ausfallsicherheit und Skalierbarkeit bietet. TIBCO liefert bei den Message Brokern vielfältige Funktionalität mit dem Schwerpunkt der Kommunikation über das Internet. Das Kernstück der ActiveEnterprise Suite ist der Informationsbus TIB. TIB/ActiveEnterprise ist eine Sammlung von einzelnen SW-Komponenten (Suite), die auf verschiedenen Betriebssystemplattformen zur Verfügung stehen, unter anderem unter WindowsNT, Sun Solaris, **HP-UX**⁷⁵ und IBM **AIX**⁷⁶.

Ebenso stehen Adapter unter den verschiedensten Plattformen (auch Host) zur Verfügung (s. Abbildung 5.1). Sollte die Plattform nicht direkt unterstützt werden, besteht im

⁷⁴ The Information Bus

⁷⁵ Hewlett Packard Unix

⁷⁶ Advanced Interactive Executive, IBM Unix-Derivat

Allgemeinen die Möglichkeit, die anzubindende Applikation Remote anzukoppeln (z. B. via Distributed API, Dateischnittstelle, usw.). Die Transportebene wird durch das eigene Produkt TIB/Rendezvous, einem Publish/Subscribe Message Bus, abgedeckt. Dabei können XML-Dokumente über HTTP über das TIBCO-eigene Protokoll RV (Rendezvous) versendet (publiziert) oder empfangen (subskribiert) werden. TIBCO stellt mit TIB/Rv Tx einen eigenen TP-Monitor zur Verfügung, mit dem langlebige Transaktionen über Anwendungen hinweg durchgeführt werden können. TIB/Rv TX kann jedoch nicht als Koordinator für 2-Phase-Commit Transaktionen fungieren, sondern sich nur als 2-Phase-Commit Ressource unter Kontrolle eines anderen TP-Monitors stellen. Die Konvertierungsebene wird durch das eigene Produkt TIB/MessageBroker abgedeckt, mit dem eine regelbasierte Konvertierung und ein inhaltsbezogenes Routing (contentbased routing) von Daten erfolgen kann. Die Konvertierung von Daten erfolgt über ein graphisches User-Interface per „Drag-and-Drop“.

Der TIB/IntegrationManager automatisiert Prozesse durch Steuerung der Reihenfolge von so genannten Events über unterschiedliche Anwendungen hinweg. Zusätzlich verwaltet der TIB/IntegrationManager den Status der einzelnen Prozesse und ermöglicht eine hierarchische und parallele Ausführung von Geschäftsprozessen sowie die Bearbeitung von Sub-Prozessen. Die Modellierung von Business-Prozessen erfolgt graphisch mit Hilfe der **UML**⁷⁷-Notation. TIB/InConcert modelliert, verwaltet und überwacht die einzelnen Tasks als Teil eines kundenorientierten Business-Prozesses, unabhängig davon, ob dieser Business-Prozess durch einen Menschen oder automatisiert ausgeführt wird. TIB/InConcert ist damit die Workflow-Komponente der ActiveEnterprise Suite. TIBCO bietet mit der TIB/Rendezvous Data Security eine Sicherheitsergänzung an. Diese bietet eine volle Unterstützung der **SSL**⁷⁸-Encryption mit 128 Bit Support. Die Einbindung von 3rd Party Sicherheitsprodukten ist ebenfalls möglich. Eine Autorisierung auf User-Ebene erfolgt über den Web-Server oder andere Frontends, eine Autorisierung auf Gruppen- bzw. Rollenebene kann z. B. über Netegrity SiteMinder erfolgen. Eine **LDAP**⁷⁹ Anbindung ist vorhanden.

Anwendungen können über zahlreich verfügbare Standard Adapter bzw. über eigen entwickelte Adapter angebunden werden. Standard Adapter zu CRM Systemen wie Clarify, Siebel und Vantive, zu SCM-Systemen wie I2 und Manugistics sowie zu ERP-Systemen wie Baan, JDEdwards, Oracle Applications, PeopleSoft und **SAP**⁸⁰ R/3 sind vorhanden. Durch die Spezialisierung von TIBCO Finance auf den Finanzbereich sind zahlreiche Adapter zu

⁷⁷ Unified Modelling Language

⁷⁸ Secure Socket Layer

⁷⁹ Light Weight Directory Access Protocol

⁸⁰ Systemanalyse Programm-Entwicklung

Finanzanwendungen wie Bloomberg, Eurex, Kondor+, Reuters Triarch und Telekurs vorhanden. Sind keine Standardadapter verfügbar, bietet TIBCO ein so genanntes TIB/Adapter **SDK**⁸¹ an, mit dem eigene Adapter entwickelt werden können. Application Server werden zum Betrieb von ActiveEnterprise nicht benötigt, können aber einfach angebunden werden. Middleware-Produkte wie z. B. MQSeries, **ORB**⁸²s, **COM**⁸³ und Oracle Queue können über eigene Adapter angebunden werden. Zu allen relevanten Datenbanken besteht die Möglichkeit des Native-Zugriffs. Darüber hinaus können Datenbanken über ODBC und **JDBC**⁸⁴ angesprochen werden.

Durch die starke Konzentration auf den Finanzbereich (TIBCO Finance) stehen Standardadapter zu wesentlichen Datenformaten des Finanzbereichs wie z. B. EDI, FIX, **IDoc**⁸⁵ und SWIFT zur Verfügung. Die Administration erfolgt über Visual Tools und unterstützt auch Scripting. Über TIB/Hawk steht ein Monitoring und Managing-Werkzeug zur Verfügung, welches einen Single-Point-of-Administration für die verteilte Struktur von ActiveEnterprise ermöglicht. Alle TIBCOProdukte haben ein so genanntes **AMI**⁸⁶, über welches die Verwaltung durch TIB/Hawk ermöglicht werden kann. Über TIB/Hawk kann ein Administrator die Anwendungsparameter, das Verhalten der Anwendung usw. überwachen und bei Bedarf oder bei Auftreten definierbarer Ereignisse eingreifen. Ein Datenlogging kann sehr einfach durch Hinzufügen eines weiteren Ausgabetrails erreicht werden. Alle Zustände und Ereignisse werden zentral in einer Datenbank abgelegt. Dadurch können mit Hilfe entsprechender Abfragen Auswertungen und Statistiken generiert werden. Weiterhin bietet auch TIB/Hawk Zugriff auf Statistiken, Log Files usw.. Die TIB/ActiveEnterprise Suite verfügt über keine integrierte Versionsverwaltung. Über Import/ Export-Funktionen im XML-Format kann aber eine Versionierung mit 3rd Party Tools erfolgen.

5.2.2 Extricity. Inc.: AllianceManager

Extricity positioniert sein EAI-Tool mit Namen AllianceManager in Richtung B2B-Integration und dabei vorwiegend auf die Unterstützung von Marktplätzen. Klassische Enterprise Application Integration innerhalb des Unternehmens ist hierfür die Voraussetzung und wird entsprechend unterstützt, stellt aber nicht den Schwerpunkt der Produktpalette dar.

⁸¹ Software Development Kit

⁸² Object Request Broker

⁸³ Component Object Model

⁸⁴ Java Database Connectivity

⁸⁵ Intermediate DOcuments

⁸⁶ Application Management Interface

AllianceManager ist von der Architektur her ein Integration Server mit Schwerpunkt B2Bintegration (Abbildung 5.2).

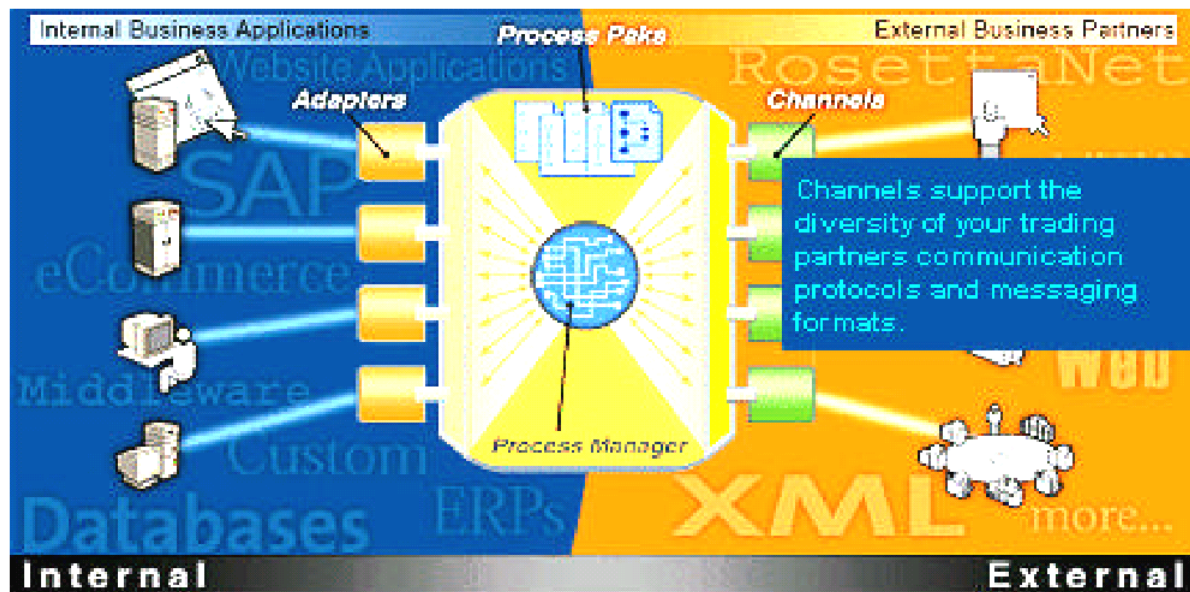


Abbildung 5.2: AllianceManager Architektur [100]

AllianceManager bietet keine eigene Transportschicht, sondern greift auf Standardanwendungen wie MQSeries, CORBA/DCOM bzw. **RMI**⁸⁷ zurück. Die Konvertierungsebene wird durch eigene Produkte, die Prozessebene durch eine eigene Process Engine abgedeckt. Durch die Integration von Application Servern erfolgt die Integration über Unternehmensgrenzen hinweg (B2B).

AllianceManager steht unter Microsoft WindowsNT und Sun Solaris zur Verfügung. Die Anbindung an andere Applikationen erfolgt über Standard API Kommunikation und ist somit generell unabhängig vom zugrundeliegenden Betriebssystem. AllianceManager ist komplett in Java realisiert. Der interne Informationsaustausch erfolgt über XML. Zur Wiederverwendbarkeit von Business Objekten kann mit dem AllianceManager eine weitere Abstraktionsebene einbezogen werden, d. h. auf Ebene der Geschäftsprozesse können Metaobjekte definiert werden, auf welche die realen Businessobjekte „gemappt“ werden. Dadurch wird eine Unabhängigkeit der Geschäftsprozesse von den anzubindenden Applikationen erreicht.

⁸⁷ Remote Method Invocation

Eine Transaktionssicherheit über mehrere Anwendungen hinweg wird nur auf Geschäftsprozessebene berücksichtigt. AllianceManager bietet die Möglichkeit, Bestätigungen anzufordern und bei Fehlschlägen der Operation ein Roll-Back zu fahren. Ein echter 2-Phase-Commit wird weder bei Einzelanwendungen, noch über mehrere Anwendungen hinweg unterstützt. AllianceManager kann auf Single- und Multi-Server Plattformen eingesetzt werden. Hinsichtlich Verschlüsselung werden Standardmechanismen wie z. B. SSL unterstützt. Eine Autorisierung wird auf User- bzw. Rollenebene unterstützt, LDAP-Unterstützung ist für zukünftige Releases vorgesehen. Anwendungen können über Wizards oder über bereits existente Adapter angebunden werden. Adapter zu SCM-Systemen wie I2, Manugistics und SAP **APO**⁸⁸ sowie zu ERP-Systemen wie Baan, SAP und Peoplesoft sind vorhanden, Pre-built Adapter zu Finanzsystemen sind nicht vorhanden. Zur Entwicklung von Adaptern steht eine **ADE**⁸⁹ zur Verfügung, zur externen Integration können die Application Server BEA WebLogic und IBM WebSphere eingebunden werden. Bezüglich Middlewareprodukte können MQSeries, CORBA bzw. DCOM sowie RMI über Adapter angebunden werden, die Anbindung von Datenbanken erfolgt über ODBC und JDBC.

AllianceManager bietet ausschließlich eine Administration über Visual Tools an. Eine Skripting-basierte Administration ist nicht möglich. Ein UML-Import zur Übernahme der Geschäftsprozesse aus entsprechenden Werkzeugen ist derzeit nicht vorgesehen. Datenlogging kann über verschiedene Logging-Stufen aktiviert werden. Ein automatisches Logging aller Dateninhalte (aus Gründen der Revision), die übertragen werden, ist dabei nicht vorgesehen und kann über Java bzw. Java-Skript ergänzt werden. Statistiken werden lediglich für den Bereich der Workflows angeboten. Es kann jederzeit verfolgt werden, wie weit ein Geschäftsprozess abgearbeitet wurde.

Bezüglich Änderungsmanagement und Versionskontrolle bietet AllianceManager eine automatische Versionierung beim Speichern des Regelwerkes. Dadurch können auch Regeln unterschiedlicher Versionsstände gleichzeitig eingesetzt werden. Produktionsversionen müssen explizit freigegeben werden. Durch eine Import/Export Funktion können versionierte Stände auch auf andere Plattformen distribuiert werden.

⁸⁸ Advanced Planner and Optimizer

⁸⁹ Adapter Development Environment

5.2.3 Microsoft: BizTalk Server 2002

Der BizTalk Server 2002 ist das zentrale Element der Microsoft Integrationsplattform. Seine Architektur ist in Abbildung 5.3 dargestellt. Im Wesentlichen gliedert sich das Produkt in zwei Teilsysteme, die zusammenarbeiten: das BizTalk Messaging und die BizTalk Orchestration.

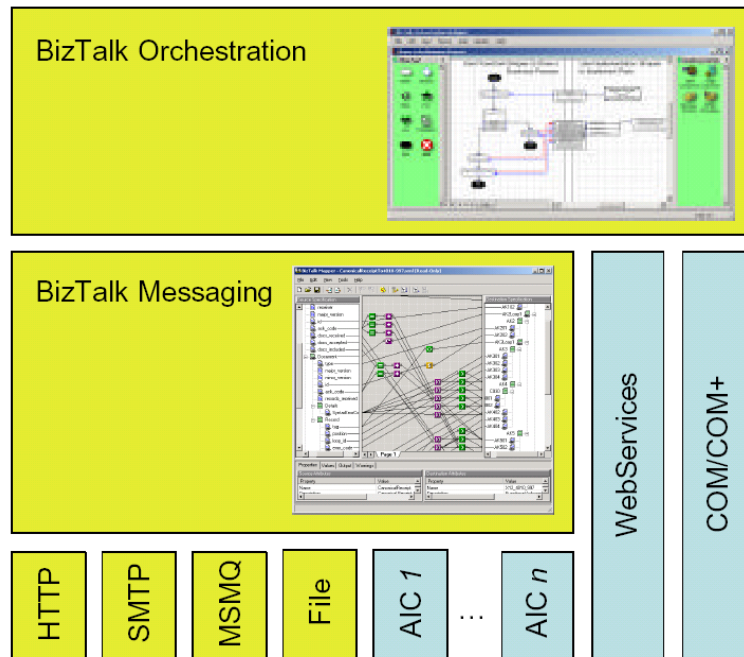


Abbildung 5.3: Architektur des Microsoft BizTalk Server 2002 [98]

Das BizTalk Messaging nimmt Nachrichten über unterschiedliche Kanäle entgegen und leitet sie entsprechend über andere Kanäle weiter. Dabei können sowohl die Dokumentformate übersetzt werden (Transformation), als auch der Zielpunkt in Abhängigkeit vom Dokumentinhalt gewählt werden (Routing). Unter den zur Verfügung stehenden Transportmechanismen findet man die üblichen Internet-Transportprotokolle, **MSMQ**⁹⁰ und Dateiaustausch. Weitere Protokolle und Technologien lassen sich über entsprechende dokumentierte Schnittstellen anbinden, so beispielsweise IBM MQseries oder auch **JMS**⁹¹. Analog können auch Applikationen angebunden werden: in diesem Fall mündet das BizTalk Messaging direkt in der Anwendung. Ein Beispiel ist SAP R/3, welches Dokumente aus dem BizTalk Messaging im IDOC Format entgegennimmt und intern weiterverarbeitet oder aber

⁹⁰ MicroSoft Message Queue

⁹¹ Java Messaging Services

Siebel 7.0, das sich ebenfalls direkt an das BizTalk Messaging ankoppelt. Es handelt sich dabei um so genannte **AIC**⁹².

Unabhängig davon, welche Transportmechanismen und Endpunkte im Einzelnen angesprochen werden, garantiert das BizTalk Messaging eine verlässliche Zustellung. Hierfür greift das System auf den Microsoft SQL Server 2000 als persistenten Speicher zurück. Eine weitere wesentliche Stärke des BizTalk Messagings ist das Transformieren von unterschiedlichen Dokumentformaten und -strukturen. Das System geht folgendermaßen vor: über generische Softwarekomponenten, den Parsern, werden die unterschiedlichen Formate eingelesen und intern in XML gewandelt. Auf Basis dieser Struktur wird die Transformation in die Zielstruktur geleistet, die wiederum als XML gehalten wird. Abschließend wird wiederum über eine generische Softwarekomponente, den Serialisierer, die Ausgabe in das spezifische Dokumentformat vorgenommen. Durch diese Architektur ist es völlig unerheblich, welche Dokumentformate im Einzelnen von den angebundenen Anwendungen erwartet werden. Intern wird als gemeinsame und leistungsfähige Basis XML verwendet.

Mit der BizTalk Orchestration lässt sich das Zusammenspiel der Systeme auf Basis von Workflows genau darstellen und umsetzen. Der BizTalk Server bringt die modellierten Abläufe direkt zur Ausführung. Eine Codierung in einer Programmiersprache entfällt. Damit lassen sich komplexe Geschäftsprozesse umsetzen und insbesondere auch den sich verändernden Anforderungen anpassen. Als grafisches Frontend dient dabei der BizTalk Orchestration Designer. An den modellierten Ablauf des Workflows werden die unterschiedlichen Systeme einheitlich über den logischen Austausch von Nachrichten angekoppelt. Für diesen Nachrichtenaustausch stehen dabei unterschiedliche Mechanismen zur Verfügung. Entweder wird auf das BizTalk Messaging mit seinen umfangreichen Möglichkeiten zurückgegriffen, oder es werden bestehende COM-Schnittstellen von Applikationen benutzt. Alternativ kann auch direkt auf Microsoft Message Queue zurückgegriffen werden. Aus dieser grafischen Darstellung wird eine XML Darstellung exportiert, die dann auf dem BizTalk Server ausgeführt wird.

Bei der Ausführung wird besonderes Augenmerk auf die Verarbeitung von Transaktionen gerichtet. Dazu zählen die bekannten Transaktionsmechanismen, die der Microsoft **DTC**⁹³ im Rahmen von COM+ anbietet und die direkt auf Datenbanktransaktionen abgebildet werden. Es besteht weiter die Möglichkeit, diese Transaktionen an bestehende Transaktionsmonitore

⁹² Application Integration Components

⁹³ Distributed Transaction Coordinator

auf anderen Plattformen anzukoppeln, wie beispielsweise den **CICS**⁹⁴ von IBM. Im Rahmen der BizTalk Server Orchestration wird häufig auch zusammenfassend von kurzlaufenden Transaktionen gesprochen. Darüber hinaus bietet die BizTalk Orchestration auch die Infrastruktur für so genannte langlaufende Transaktionen, mit denen Geschäftsprozesse abgebildet werden können, die über Tage, Wochen oder Monate ablaufen. Laufende Workflows innerhalb der BizTalk Orchestration können an diesen unterschiedlichen Transaktionen teilnehmen oder diese initiieren. Ihr detailliertes Verhalten bei fehlgeschlagenen Transaktionen kann grafisch modelliert werden.

5.2.4 Vitria: BusinessWare

Vitria positioniert sein EAI-Tool mit Namen BusinessWare als real-time Werkzeug zur Integration, Kommunikation, Analyse und Automation für e-Business. Mit BusinessWare wird die gesamte Palette der Integration von A2A bis B2B Integration mit einem Werkzeug abgedeckt. Darüber hinaus liegt ein weiterer Schwerpunkt von BusinessWare in der RosettaNet-Integration. Alle BusinessWare-Komponenten benutzen das gleiche graphische Userinterface und sind somit auch aus Anwendersicht vollständig integriert.

BusinessWare ist als Message Broker mit konsequenter Umsetzung eines Publish/Subscribe Verfahrens realisiert. Dies bedeutet die Realisierung eines föderativen Konzeptes (Multihub) ohne zentrale Komponenten, was Vorteile bezüglich Ausfallsicherheit und Skalierbarkeit bietet. BusinessWare besteht aus folgenden Komponenten (siehe auch Abbildung 5.4):

- BusinessWare Automator (Definition der Geschäftsprozesse)
- BusinessWare Analyzer (real-time Analyse Tool)
- BusinessWare Communicator (Messaging Komponente, CORBA/IIOP⁹⁵ compliant)
- BusinessWare Connectors and Transformers (Application Integration Komponenten)
- BusinessWare Server (stellt die Host-Umgebung für vorgenannte run-time Module zur Verfügung)

⁹⁴ Customer Information Control System

⁹⁵ Internet Inter-ORB Protocol

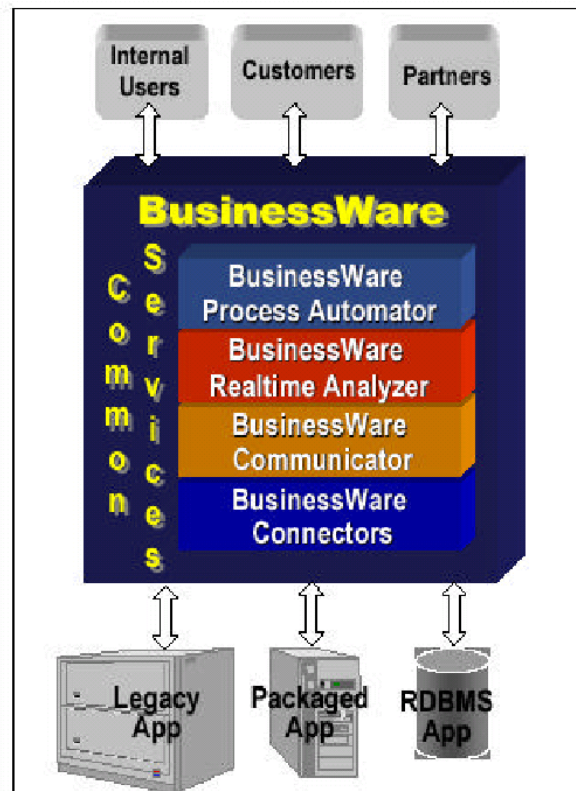


Abbildung 5.4: BusinessWare Architektur [100]

Businessware deckt somit die Transport-, Konvertierungs- und Prozessebene durch eigene Produkte ab. Es ist anzumerken, dass alle Komponenten von BusinessWare von Anfang an als vollständig integrierte Lösung konzipiert waren und entsprechend durch Vitria realisiert wurden. Im Gegensatz zu vielen Mitbewerbern bietet Vitria somit eine homogene und vollständig integrierte Lösung an.

BusinessWare steht unter WindowsNT, Sun Solaris, HP-UX und IBM AIX zur Verfügung, hingegen gibt es die Konnektoren auf den verschiedensten Plattformen (auch Host). Sollte die Plattform nicht direkt unterstützt werden, besteht im Allgemeinen die Möglichkeit, die anzubindende Applikation remote anzubinden (z. B. via Distributed API, Dateischnittstelle, usw.). BusinessWare basiert komplett auf dem CORBA-Standard, wobei eine eigene ORB Implementierung genutzt wird. Als Transportmechanismus wird ein Publish/Subscribe Verfahren benutzt (BusinessWare Communicator). BusinessWare stellt einen eigenen Transaktionsmanager zur Verfügung. Dieser bietet Transaktionssicherheit für alle BusinessWare-Komponenten. Er unterstützt 2-Phase-Commit und so genannte „Compensating Transactions“ und erlaubt die Einbindung von 1-Phase-Commit und 2-Phase-Commit Ressourcen in einem über mehrere Applikationen übergreifenden Geschäftsprozess. Die BusinessWare Konnektoren bestehen aus jeweils drei Teilen. Der Treiber stellt die Verbindung zur jeweiligen anzubindenden Applikation dar (z. B. Flat-File, API, usw.), der

Transformer enthält die Transformationsregeln und die Business-Rules, die über die graphische Oberfläche per Drag-and-Drop modelliert werden können. Der letzte Teil publiziert bzw. subskribiert die Informationen auf bzw. von einem so genannten Channel.

Der BusinessWare Automator ist die Business-Prozess-Komponente von BusinessWare. Hier können z. B. durch einen Business Analysten direkt die Business-Prozesse graphisch modelliert und mit den Connection-Modellen (Konnektoren mit Transformationslogik und Business-Rules) verbunden werden. Für die graphische Bearbeitung wird die Notation nach UML genutzt. Kommunikation über öffentliche Medien (z. B. Internet) wird per SSL verschlüsselt. Zu diesem Zweck enthält BusinessWare BSAFE Verschlüsselung oder Security Protocol Software von **RSA**⁹⁶ Security. Eine Autorisierung erfolgt über den BusinessWare User Manager, der eine Zugangskontrolle über **ACL**⁹⁷ ermöglicht. BusinessWare kann LDAP-Services einbinden, hat aber den LDAP-Standard um eigene Funktionen erweitert. Anwendungen können über ein SDK oder über vorhandene Konnektoren (über 90 Stück, stetig wachsende Anzahl) angebunden werden. Standard-Konnektoren zu CRM-Systemen wie Clarify, Siebel und Vantive, zu SCM-Systemen wie SAP APO und ERP-Systemen wie Oracle, PeopleSoft, SAP R/3 und SAS/DataWarehouse sind vorhanden. Zur Entwicklung von Konnektoren steht ein SDK zur Verfügung. Die Entwicklung von Konnektoren ist stark kundengetrieben, d. h. bei kommerzieller Attraktivität ist ein entsprechender Support seitens Vitria möglich.

Alle Modelle sowie die Metadatendefinition können aus/in BusinessWare exportiert/importiert werden. Modelle in BusinessWare basieren auf UML. Durch die starke CORBA-Bindung wird darüber hinaus **IDL**⁹⁸ unterstützt. BusinessWare benötigt zur Enterprise Application Integration keinen Application Server. Im Rahmen von Kundenprojekten wurden aber bereits BEA's WebLogic und IBM's WebSphere angebunden. Die Anbindung von Datenbanken erfolgt über native Zugriffe (DB2, Informix, Oracle, SQL Server, Sybase) bzw. JDBC. Als Datenquellen bzw. Datenformate werden standardmäßig **CSV**⁹⁹, Flatfile, XML, und EDI unterstützt. Die Anbindung von **EBCDIC**¹⁰⁰ wurde für einen Kunden realisiert und ist laut Vitria intern verfügbar.

⁹⁶ Ron Rivest, Adi Shamir, Leonard Adelman

⁹⁷ Access Control List

⁹⁸ Interface Definition Language

⁹⁹ Comma Separated Values

¹⁰⁰ Extended Binary-Coded Decimal Interchange Code

Die Administration von BusinessWare erfolgt über Visual Tools. Da BusinessWare komplett auf dem CORBA-Standard basiert, können alle Komponenten weltweit verteilt eingesetzt, aber über einen Single-Point-of-Administration zentral administriert werden. Darüber hinaus ist BusinessWare für den Einsatz in **HA**¹⁰¹-Clustern ausgelegt und ein UML-Import/Export zu Rational Rose ist standardmäßig vorhanden. Das Regelwerk kann zur Laufzeit upgedated werden; laufende Prozesse werden mit dem „alten“ Regelwerk beendet, neue Prozesse greifen auf das neue Regelwerk zu. Über den BusinessWare Analyzer können Geschäftsprozesse und Daten zur Laufzeit in real-time überwacht und Statistiken erstellt werden. Datenlogging ist für jede Komponente in fünf unterschiedlichen Leveln möglich. Es können Default-Logs und Individual-Logs konfiguriert werden. Alle Prozesse und ihr Status können in einer Datenbank geloggt werden (Audit-Trails).

BusinessWare speichert statische Informationen (wie z. B. Mapping und Regelwerk) in einer Datenbank, schnelllebige Daten wie aktuelle Events werden in einem hochperformanten Dateisystem gespeichert. Dies ermöglicht BusinessWare zum einen hohen Datendurchsatz, zum anderen kann im Fall eines System-Crashes automatisch auf dem letzten Zustand aufgesetzt und die Verarbeitung fortgesetzt werden. BusinessWare bietet eine integrierte Versionsverwaltung auf Komponentenebene. Dadurch können auch Regeln unterschiedlicher Versionsstände gleichzeitig eingesetzt werden. Einzelne Versionen (z. B. Produktionsversionen) können für externen Zugriff gesperrt (locked) werden. Durch eine Import/Export Funktion können versionierte Stände auf andere Plattformen distribuiert oder unter Versionskontrolle externer Versionsverwaltungswerkzeuge (z. B. RCS, **PVCS**¹⁰², Continuous, usw.) gestellt werden. Darüber hinaus kann der Zugriff auf einzelnen Versionen durch den Autorisierungsmechanismus (ACL) festgelegt werden.

5.2.5 Data Mirror: Constellar Hub

Data Mirror positioniert sein EAI-Tool Constellar Hub mit Schwerpunkt auf die Datenintegration und Transformation in heterogenen Systemumgebungen. Die Transportschicht wird bei Constellar Hub durch 3rd Party Produkte wie MQSeries oder Oracle Queue, die Konvertierungsebene wird durch eigene Produkte abgedeckt. Die Architektur ist in Abbildung 5.5 dargestellt.

¹⁰¹ High Availability

¹⁰² Polytron Version Control System

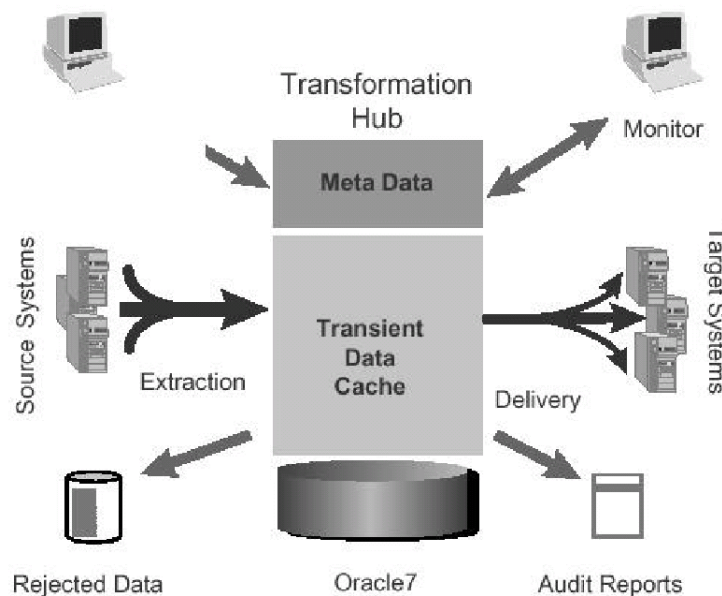


Abbildung 5.5: Architektur von Constellar Hub [100]

Constellar Hub steht unter AIX, HP-UX, WindowsNT und Sun Solaris zur Verfügung. Die Anbindung der gängigen Source-/Target-Plattformen ist ebenfalls gewährleistet. Konnektoren und Datentransformationen werden über eine eigene Skriptsprache (**TCL**¹⁰³) realisiert. TCL-Skripte können in **PCL**¹⁰⁴ oder C kompiliert werden. Darüber hinaus können komplexe Transformationen auch durch C- oder Java-Programme, aufgerufen aus entsprechenden TCL-Skripten, realisiert werden. Die Architektur von Constellar Hub unterstützt derzeit keine **EJB**¹⁰⁵s oder ORBs und ist auf Oracle spezialisiert. Constellar Hub bietet keinen Transaktionsmonitor. Mehrere Daten können als Paket verschickt werden. Dadurch kann sowohl eine Reihenfolgegetreue als auch ein „Alles oder Nichts“ Prinzip abgebildet werden. Eine Transaktionssicherheit im Sinne eines 2-Phase-Commit wird jedoch nicht unterstützt.

Die Kommunikation über öffentliche Medien erfolgt über Standardmechanismen wie z. B. SSL. Constellar Hub verwendet keine eigenen Verschlüsselungsmechanismen. Eine Autorisierung wird auf User- bzw. Rollenebene unterstützt, eine LDAP-Unterstützung ist ebenfalls vorhanden, Pre-Built-Adapter zu CRM- und SCM-Systemen jedoch nur eingeschränkt. Zu den wesentlichen ERP-Systemen wie Baan, **JDE**¹⁰⁶, SAP und Peoplesoft sind Adapter vorhanden. Adapter zu Finanzsystemen (Bloomberg, Devon, Kondor und Reuters Triarch) wurden im Rahmen von Kundenprojekten realisiert. Die Architektur von

¹⁰³ Tool Command Language

¹⁰⁴ Programmable Command Language

¹⁰⁵ Enterprise Java Beans

¹⁰⁶ JD Edwards

Constellar Hub benötigt keinen Application Server. Bezüglich Middlewareprodukten können MQSeries, Oracle Queue sowie RMI und RPC genutzt werden. Alle relevanten Datenbanken können nativ angesprochen werden, zusätzlich stehen ODBC und JDBC zur Verfügung. Durch die sehr starke Anbindung der eigenen Architektur an Oracle ist hier eine gute Integration gegeben. Constellar bietet vielfältige Datenformate auch aus dem Finanzbereich an. Über die eigene Transformation Definition Language können zusätzlich komplexe Transformationen realisiert werden.

Constellar Hub kann sowohl über Visual Tools als auch über Skripte administriert werden. Ein UML-Import zur Übernahme der Geschäftsprozesse aus entsprechenden Werkzeugen ist nicht möglich. Datenlogging kann über verschiedene Logging-Stufen aktiviert werden. Ein automatisches Logging aller Dateninhalte (aus Gründen der Revision), die übertragen werden, ist dabei allerdings nicht automatisch vorgesehen und muss durch Programmierung ergänzt werden. Statistiken werden für Performancemessungen und Änderungen angeboten. Zur Versionierung bietet Constellar Hub lediglich Import/Export Funktionen an. Die Versionierung muss in einem externen Werkzeug erfolgen.

5.2.6 CrossWorlds: CrossWorlds

CrossWorlds positioniert sich mit seinem gleichnamigen Produkt als „near real-time“ Integration Server mit Schwerpunkt auf XML. Zentraler Bestandteil der CrossWorlds Architektur ist der Interchange Server. Der Interchange Server ist in Java realisiert.

Integrationsprozesse werden bei CrossWorlds als Collaboration bezeichnet. Wird eine Collaboration gestartet, wird diese an die Applikation „gebunden“ (s. auch Abbildung 5.6). Bezüglich der Datentransformation führt CrossWorlds einen zusätzlichen Abstraktionslayer ein und arbeitet mit so genannten „generischen Businessobjekten“. Dadurch erreicht CrossWorlds eine Unabhängigkeit der anzubindenden Applikationen und Wiederverwendbarkeit der generischen Businessobjekte (Abbildung 5.7).

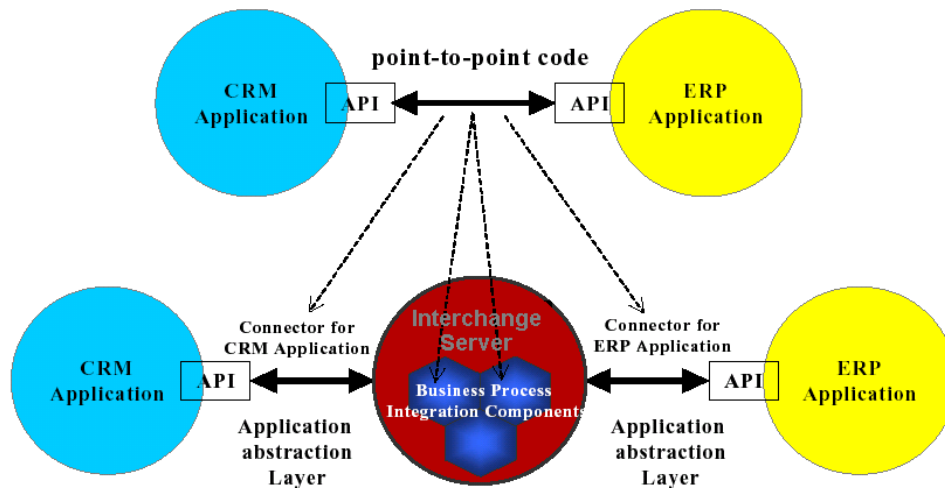


Abbildung 5.6: Interchange Server als Bestandteil der CrossWorlds Architektur [100]

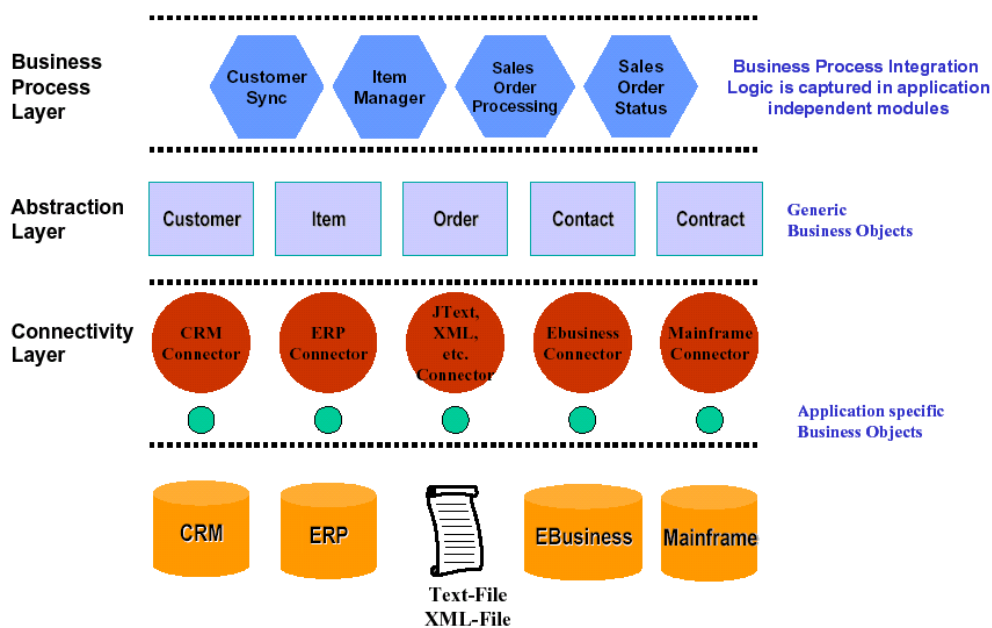


Abbildung 5.7: Mapping von realen Businessobjekten auf generische Businessobjekte [100]

Es existieren eine Vielzahl vordefinierter generischer Businessobjekte für Industrie und Telekommunikation (stetig wachsende Anzahl). Generische Businessobjekte können auch selbst erstellt bzw. importiert/exportiert werden. Für den reinen Datentransport konzentriert CrossWorlds sich auf 3rd Party Produkte wie z. B. MQSeries oder CORBA/IIOP. Die Konvertierungsebene wird durch den Cross-Worlds Map Designer, die Prozessebene durch den CrossWorlds Process Designer (beides eigene Produkte) abgedeckt. Durch Nutzung der so genannten Collaborations mit dem Open Access Framework ist auch eine externe Integration (B2B) beispielsweise über das Internet möglich. CrossWorlds ist eine Java-

Applikation und steht unter Sun Solaris und WindowsNT zur Verfügung. Hinsichtlich der Anbindbarkeit von Quell-/Zielsystemen besteht keine Einschränkung. Sollte die Plattform nicht direkt unterstützt werden, kann die Applikation auch remote angebunden werden, d. h. via Distributed API, Dateischnittstelle, usw..

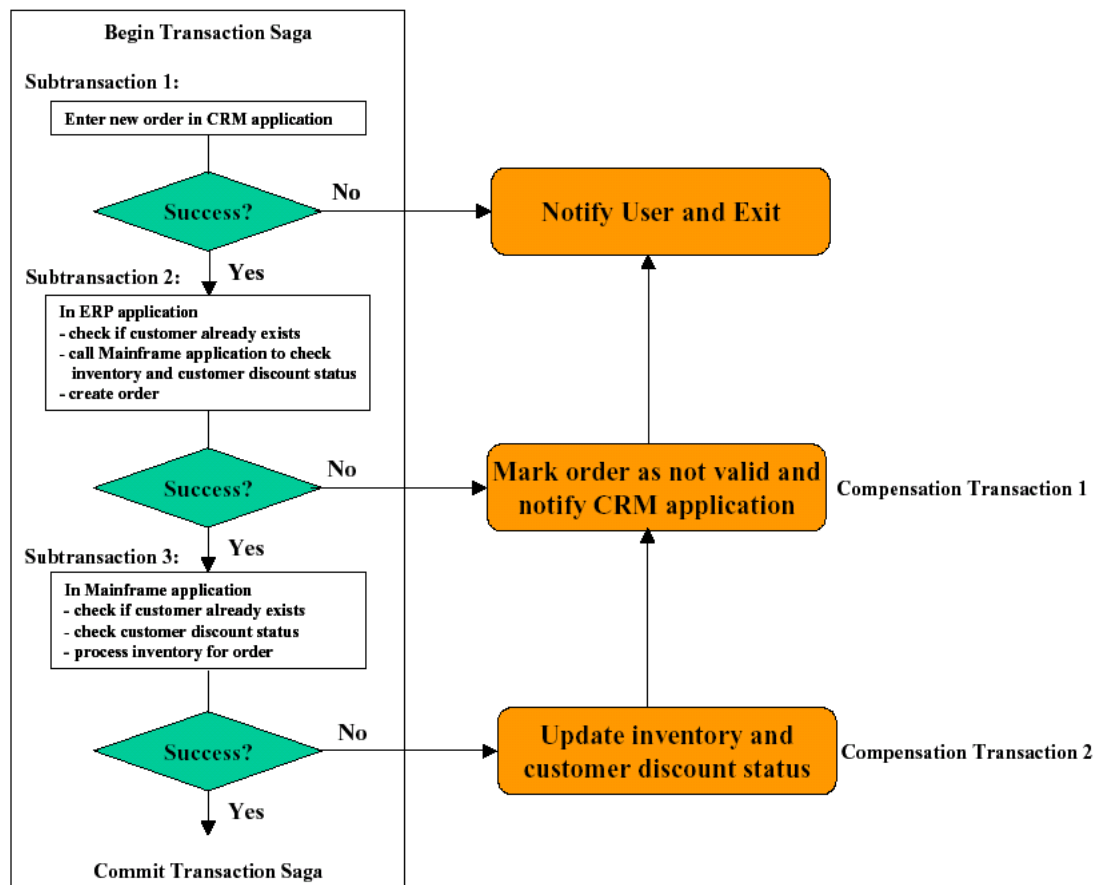


Abbildung 5.8: Transaktionen bei CrossWorlds für langlebige Prozesse [100]

CrossWorlds stellt Transaktionen für so genannte langlebige Prozesse nach dem **CHAT**¹⁰⁷-Modell sicher. Nach diesem Modell werden verteilte Transaktionen in eine Kette von Einzeltransaktionen aufgebrochen. Dabei fungiert die Kette wiederum als Transaktion. Sollte eine Einzeltransaktion fehlschlagen, kann ein Roll-Back in Form einer "Compensating Transaction" auch über mehrere Anwendungen hinweg durchgeführt werden. CrossWorlds speichert die Ausgangswerte solange zwischen, bis alle Anwendungen „committed“ haben (siehe auch Abbildung 5.8). Ein echter 2-Phase-Commit über mehrere Anwendungen hinweg wird dabei nicht sichergestellt. CrossWorlds stellt die Reihenfolge (Sequencing) der Einzeltransaktionen sicher, was aber negative Einflüsse auf die Performance haben kann.

¹⁰⁷ CrossWorlds Hybrid Asynchronous Transaction Model

Der Einsatz von CrossWorlds auf Single- und Multi-Server Plattformen und HA-Clustern ist möglich. CrossWorlds unterstützt derzeit ein statisches Load-Balancing, ein dynamisches Load-Balancing ist für zukünftige Releases geplant.

CrossWorlds enthält keine eigene Verschlüsselungsmechanismen, sondern greift auf Verschlüsselungsmechanismen der darunter liegenden Transportschicht (z. B. SSL, MQSeries) zurück. Eine Autorisierung erfolgt ausschließlich auf User-Ebene, LDAP-Unterstützung besteht derzeit nicht, ist aber als Produktmerkmal für zukünftige Releases geplant. Anwendungen können über vorhandene Standard-Konnektoren bzw. über eigen entwickelte Konnektoren integriert werden. Standard-Konnektoren zu CRM-Systemen wie Clarify, Siebel und Vantive, zu SCM-Systemen wie JDE, Manugistics und SAP APO sowie zu ERP-Systemen wie Baan, JDE, PeopleSoft, SAP R/3 und Oracle stehen zur Verfügung, pre-built Konnektoren zu Finanzsystemen existieren nicht.

CrossWorlds bietet so genannte „Application Connectors“ und „Technology Connectors“ an. Während die Application Connectors die Anbindung an Standardanwendungen zur Verfügung stellen, unterstützen die Technology Connectors die Konnektivität zu Datenbanken, Datenformaten (z. B. JText) oder Zugriffsmethoden (z. B. JDBC). Die Eigenentwicklung von Konnektoren kann unter Java bzw. C++ erfolgen. Bei der Datenbank-Connectivity können über einen separat erhältlichen Datenbank-Wizard auch vorhandene Datenbankstrukturen ausgelesen und für das Mapping verwendet werden. CrossWorlds unterstützt ein Cross-Referencing von Datenwerten (z. B. Abbildung von Kurzbezeichnungen in Langbezeichnungen). Die Administration erfolgt über Visual Tools und unterstützt auch Scripting. Ein Monitoring der einzelnen Verarbeitungsschritte ist möglich. Es werden verschiedene Logging-Level unterstützt, bei denen auch die übertragenen Daten mitgeloggt werden können. Ein spezifisches Logging der Daten für Revisions-Zwecke ist standardmäßig nicht vorgesehen, kann aber einfach über zusätzliche Ausgabepfade realisiert werden. Im Fehlerfall kann eine Signalisierung über Events und/oder e-Mail erfolgen. Der Cross-Worlds System Manager unterstützt ein Account-Management. Es können Statistiken über abgeschlossene Workflows und Performance angezeigt werden. Eine Versionierung verschiedener Collaborations erfolgt über Namenskonvention. Eine integrierte Versionsverwaltung ist nicht vorhanden.

5.2.7 SeeBeyond: e*Gate / e*Xchange

SeeBeyond wurde 1991 als STC mit der Vision gegründet, sämtliche Systeme, Applikationen, Geschäftspartner, Lieferanten und Kunden des Unternehmens zu integrieren, wobei der dynamische Informationsaustausch oberstes Gebot war. SeeBeyond bietet mit e*Gate eine Integrationsplattform, die unternehmensweit oder auch weltweit distribuiert eingesetzt werden kann. Version 4.0 ist die vierte Generation von Data Gate, welches bei mehr als 1.500 Kunden weltweit implementiert wurde. SeeBeyond positioniert e*Gate innerhalb des ebusiness Integration Marktes, was für SeeBeyond die Schnittstelle zwischen EAI und B2B Integration bedeutet.

Die e*Gate Plattform ist von ihrer Architektur her ein Message Broker (Network-Centric). Die unbegrenzt skalierbare, netzwerkzentrierte Architektur sorgt auf allen Host-Systemen im Netzwerk für den dynamischen Einsatz der Komponenten und der Business Logik. Alle Komponenten sind Multi-Thread-fähig. Somit wird die **CPU**¹⁰⁸ eines jeden Host-Systems voll genutzt, um das Antwortzeitverhalten mit Realtime-Geschwindigkeit sicherzustellen, und zwar unabhängig davon, ob täglich nur einige wenige Transaktionen oder mehrere Millionen Transaktionen zu verarbeiten sind. Die Konfigurationsdaten werden zentral in einem Repository gespeichert.

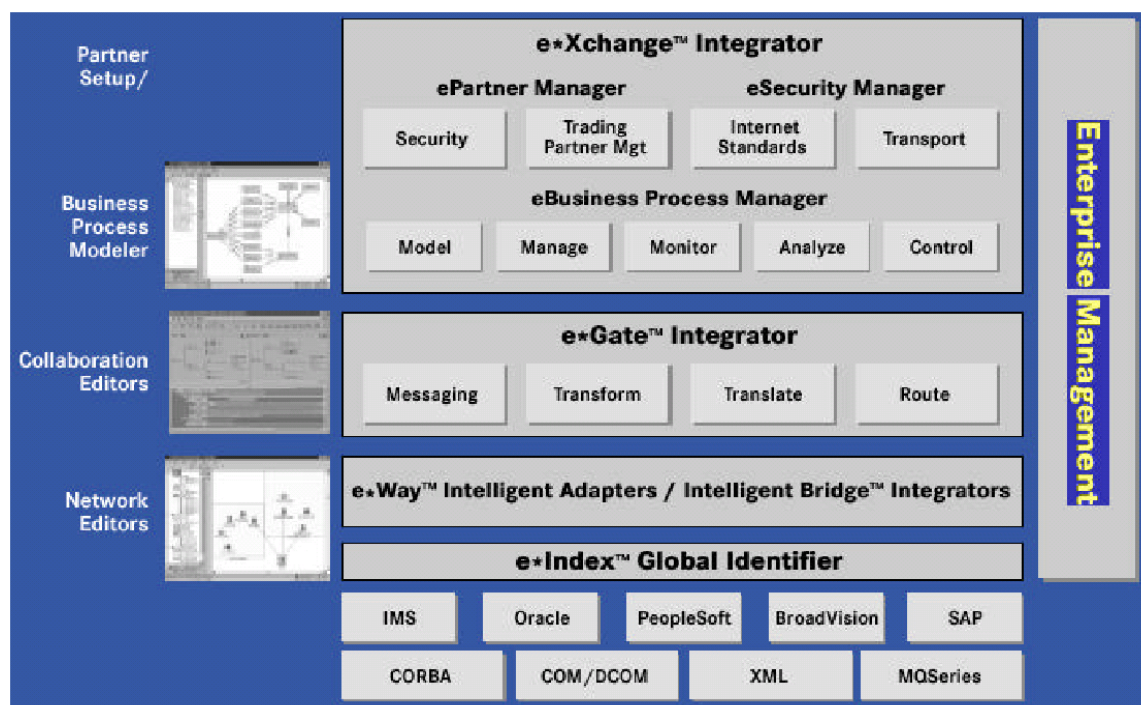


Abbildung 5.9: Architektur von e*Gate / e*Xchange [100]

¹⁰⁸ Central Processing Unit

Die Transportschicht wird durch CORBA abgedeckt, alternativ kann auch z. B. MQSeries eingesetzt werden. Die Konvertierungs- und Prozessebene werden durch eigene Produkte mit Namen e*Gate und e*Xchange abgedeckt (siehe auch Abbildung 5.9).

Integrationsprozesse werden bei SeeBeyond als Collaborations bezeichnet. Collaborations beinhalten die Business-Logik, die Transformationslogik, die Routing-Regeln und die Benachrichtigungs- und Alarmierungsregeln. Um die volle Funktionalität zu gewährleisten, auch wenn das Repository kurzzeitig nicht zur Verfügung steht, isoliert der Control Broker die Laufzeitumgebung von der Registry. Das in e*Gate integrierte pub/sub-Modell ordnet den einzelnen Business-Events die Collaborations zu und unterstützt die Verknüpfungstypen n:1, 1:n und n:n. Darüber hinaus können die pub/sub-Verknüpfungen jederzeit dynamisch modifiziert werden, ohne dabei die Business-Logik zu verändern. e*Gate stellt Transaktionen für langlebige Business-Prozesse zur Verfügung, unterstützt jedoch kein 2-Phase-Commit über mehrere Anwendungen hinweg. e*Xchange eSecurity Manager sorgt für die Sicherheit der Kommunikation in öffentlichen Netzen.

Mit der **PKI**¹⁰⁹ wird die Verschlüsselung, Prüfung der Integrität des Inhalts, Prüfung der Zugriffsberechtigung und Autorisierung sichergestellt. Mit Hilfe von rollenbasierten Profilen wird das gesamte System vor dem Zugriff Unbefugter geschützt. Anwendungen können über vorhandene bzw. selbst entwickelte Adapter angebunden werden, Standardadapter stehen für PeopleSoft, SAP R/3, Siebel, Vantive und BroadVision zur Verfügung sowie einige wenige Adapter für ausgewählte Finanzanwendungen.

Das wesentliche Konzept, um e-Business-Lösungen zu implementieren, besteht darin, die vorhandenen Systeme zu betrachten und neue Technologien oder Services auf einem gemeinsamen Netzwerk bereitzustellen, welche von diesen Applikationen genutzt werden können. e*Gate stellt hierfür den so genannten „Intelligent Queuing Layer“ und den „Application Connectivity Layer“ zur Verfügung. Intelligent Queues stellen die Interprozess-Kommunikation auf Basis des eigenen Queuing Systems, aber auch auf Basis MQSeries zur Verfügung. Die eigentlichen Adapter werden bei SeeBeyond als e*Ways bezeichnet. Die e*Ways Adapter sind für die Verbindung zu den Applikationen zuständig und unterstützen Datenmanipulationen wie „Business Collaboration“, Transformations-Logik und pub/sub-Beziehungen. Die e*Way Adapter arbeiten im Multi-Thread-Verfahren und stellen damit dem Unternehmen verteilte Verarbeitungskapazitäten von höchster Performance zur Verfügung. Damit ist eine absolute Flexibilität im Produktionsbereich, sowie eine ausgewogene

¹⁰⁹ Public Key Infrastructure

Verteilung der Arbeitslasten sichergestellt. SeeBeyond hat über 500 e*Way Adapter entwickelt bzw. Kunden bei deren Entwicklung unterstützt. Diese Adapter sind in folgende Gruppen unterteilt:

- Datenbankzugriff, Retrieval und Update (z. B. Oracle, Sybase und ODBC)
- Finanzdienstleistungsbereich / Management (SWIFT)
- Middleware (z. B. COM/DCOM, CORBA, EJB, IBM MQSeries)
- Kommunikationsprotokolle (z. B. **TCP/IP**¹¹⁰, **FTP**¹¹¹, HTTP(S), **SMTP**¹¹²)
- Altsysteme und Datenspeichersysteme (z. B. CICS, **IMS**¹¹³, DB2)
- Generic e*Way Extension Kit - individuelle Anbindung von Applikationen.

Die Administration von e*Gate erfolgt über Visual Tools mit einer Single-Point-of-Administration. Ein Monitoring der einzelnen Verarbeitungsschritte ist möglich. Daten können einfach über eine weitere Ausgabe geloggt werden. Im Fehlerfall kann eine Alarmierung über e-Mail, Events oder SNMP erfolgen.

5.2.8 New Era of Networks (NEON): e-Biz Integrator

NEON ist einer der ersten Hersteller, die EAI-Tools angeboten haben. Es besteht ein Patent auf die NEON Rules and Formatter Engine, die auch in anderen EAI-Tools (z. B. !Candle ROMA) eingesetzt wird. NEON hat eine sehr enge Partnerschaft mit IBM und bietet mit dem e-Biz Integrator (Abbildung 5.10) eine starke Integrationsplattform auf Basis MQSeries als Transportsystem, weitere werden unterstützt.

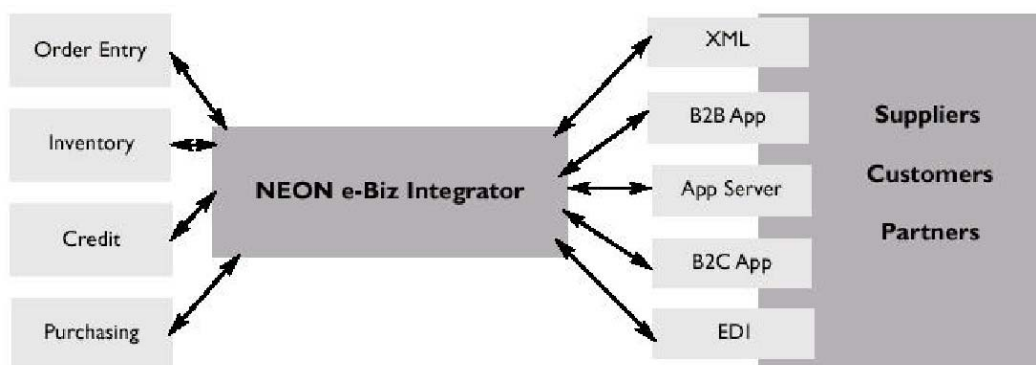


Abbildung 5.10: e-Biz Integrator [100]

¹¹⁰ Transmission Control Protocol/Internet Protocol

¹¹¹ File Transfer Protocol

¹¹² Simple Mail Transfer Protocol

¹¹³ Information Management System

Der e-Biz Integrator ist ein Integrationsserver, welcher Applikationen in einer zentralen Hub and Spoke Architektur integriert. Alternativ zu MQSeries können derzeit TIB/RV und DCOM als Transportsysteme verwendet werden. In der aktuellen Version wird zusätzlich ein eigen entwickeltes Transportsystem **OTI**¹¹⁴ unterstützt, über welches auch andere Transportsysteme angebunden werden sollen.

e-Biz Integrator arbeitet intern mit Metaobjekten, d. h. Businessobjekte werden für das Mapping in ein internes Format und von dort in das Zielformat abgebildet. Das Routing von Daten kann inhaltsbasiert erfolgen. Die Prozessebene wird durch den NEON Process-Server abgedeckt, mit dem die Integrationsprozesse graphisch erstellt und gepflegt werden können. Aus Performancegründen ist der e-Biz Integrator eine C++-Applikation. Er steht unter allen wesentlichen Betriebssystemen zur Verfügung. Wird als Transportsystem MQSeries verwendet, ist eine Transaktionssicherheit mit 2-Phase-Commit über mehrere Anwendungen hinweg nicht möglich. Die Reihenfolgetreue von Datensätzen wird durch das Transportsystem (MQSeries) sichergestellt.

Bezüglich der Verschlüsselung auf der Transportebene können 3rd Party Produkte eingebunden werden. e-Biz Integrator bietet keine eigenen Autorisierungsmechanismen. Da der e-Biz Integrator aber als Repository eine Datenbank sowie als Transportsystem MQSeries verwendet, greifen die dortigen Autorisierungsmechanismen, d. h. Zugriffe auf das Repository erfordern ein entsprechendes Datenbank-Login, Zugriffe auf MQSeries werden über ACLs verwaltet. Alternativ dazu kann mit NEON Secure ein Sicherheits- und Autorisierungsprodukt aus dem eigenen Haus eingebunden werden. NEON Secure arbeitet mit Eigenschaften und Regeln, die entsprechend erfüllt werden müssen, eine LDAP-Server Anbindung ist unter WindowsNT verfügbar.

Der e-Biz Integrator stellt Adapter zu Siebel, I2, JDEdwards, Oracle Applications, PeopleSoft, SAP R/3 und SAS/Datawarehouse zur Verfügung. Darüber hinaus bietet NEON auch Connectivity zu SWIFT. Im Rahmen von Kundenprojekten wurden verschiedene Adapter für Finanzanwendungen realisiert. NEON bietet neben seinen funktionalen Adaptern zu Standardanwendungen auch sogenannte Technologieadapter wie zu **COBOL**¹¹⁵ CopyBooks, XML, SWIFT usw.. Insbesondere im Hinblick auf die Connectivity zum Mainframe erfolgt die Konvertierung von gepacktem EBCDIC transparent für den Nutzer. Datenbanken können über ODBC angebunden werden. Zur Anbindung von Inhouse-Anwendungen steht ein

¹¹⁴ Open Transport Interface

¹¹⁵ Common Business Orientated Language

eigenes ADK zur Verfügung. Die Administration erfolgt über Visual Tools. Neue Integrationsprozesse können dabei zur Laufzeit distributiert und aktiviert werden. Es werden verschiedene Logging-Level unterstützt. Ein Data-Logging kann einfach durch einen weiteren Ausgabepfad bzw. zu Audit-Trails in eine Datenbank realisiert werden. Der e-Biz Integrator verfügt über keine integrierte Versionsverwaltung. Eine Versionierung kann über Import/Export-Funktionen unter Einbindung eines externen Versionsverwaltungs-Tools erfolgen.

5.2.9 BEA Systems: eLink / Tuxedo / WebLogic

BEA unterstützt mit seiner Produktserie real-time- und e-Business-Anwendungen. Die Transport-, Konvertierungs- und Prozessebene werden durch eigene Produkte abgedeckt. Durch die vollständige Integration des WebLogic Application Servers ist auch eine externe Integration, z. B. über das Web, möglich. Die Userinterfaces der Produkte sind GUI-basiert. Die e-Link Plattform kann von ihrer Architektur her der Middleware-Integration zugeordnet werden. Der Kern der BEA-Tools bildet die eLink Plattform, durch sie wird die Transportschicht völlig abgekapselt. Der Transport kann über Queuing Systeme wie z. B. BEA, MessageQ und MQSeries, aber auch über CORBA bzw. DCOM erfolgen.

BEA unterstützt mit seiner Produktreihe alle wesentlichen Plattformen. Insbesondere durch die gute Host-Unterstützung ist BEA für Finanzunternehmen interessant. Die Architektur der BEA Produkte ist objektorientiert. Es werden EJBs und CORBA bzw. DCOM unterstützt. Transaktionssicherheit (2-Phase-Commit über mehrere Anwendungen hinweg) wird durch BEA Tuxedo gewährleistet. Tuxedo setzt auf eLink auf und verdeckt ebenfalls die Transportschicht. Bezüglich der Skalierbarkeit werden durch BEA Single-Server und Multi-Server Umgebungen sowie HA-Cluster unterstützt. Ein Load Balancing wird sowohl durch Tuxedo als auch durch WebLogic unterstützt.

Die e-Link Plattform enthält keine eigenen Verschlüsselungsmechanismen, sondern greift auf Verschlüsselungsmechanismen der darunter liegenden Transportschicht (z. B. SSL) zurück. Alternativ können 3rd Party Produkte eingebunden werden. Eine Autorisierung ist auf User-, Gruppen- und Rollenebene möglich, ebenfalls besteht eine LDAP-Unterstützung. Die eLink-Plattform bietet ein so genanntes Application Adaptor Framework, über welches Standard- und Inhouse-Applikationen angebunden werden können. Standard-Adapter zu CRM-Systemen wie Clarify, Maximizer, Siebel, Vantive und zu ERP-Systemen wie Oracle Applications, PeopleSoft und SAP R/3 stehen zur Verfügung (Abbildung 5.11). Im

Allgemeinen müssen die Standardadapter auf den jeweiligen Anwendungsfall angepasst werden (Customizing).

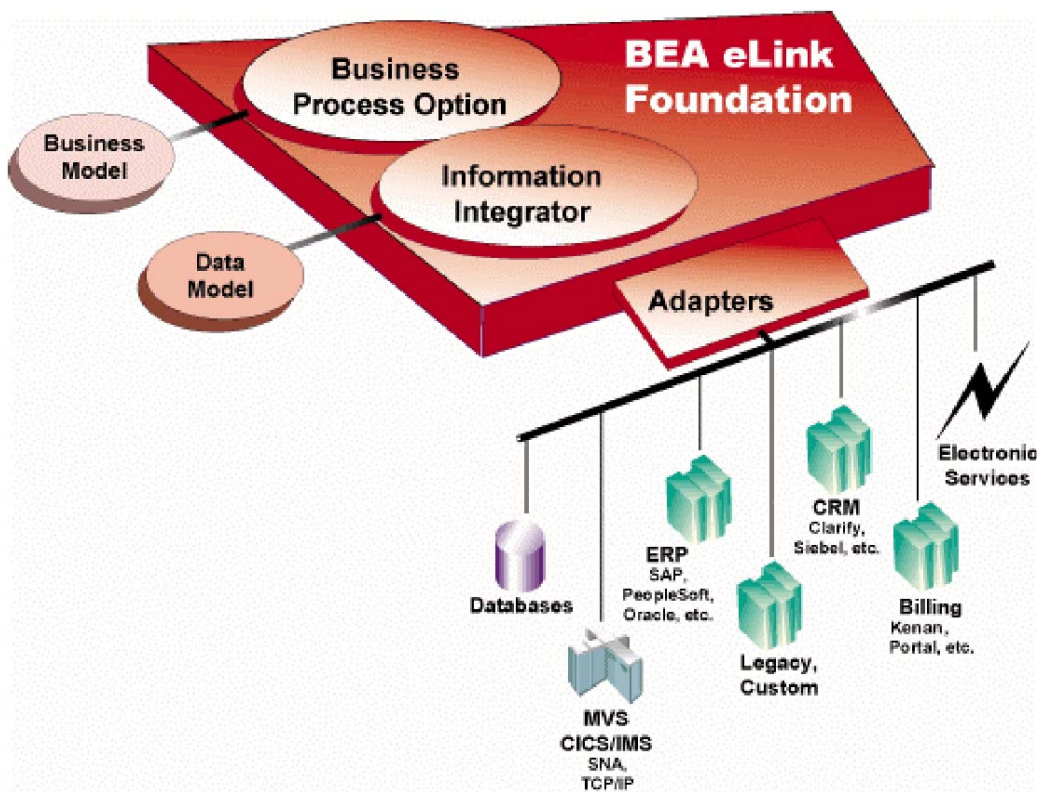


Abbildung 5.11: eLink Plattform [100]

Neben den Standard-Adaptoren können über ein ADK auch spezielle Adapter für Inhouse-Applikationen realisiert werden. Über die Data Integration Option wird ein Meta-Datenmodell modelliert, über welches ein Datenmapping von einem so genannten Businessobjekt der Applikation A in ein Businessobjekt der Applikation B auf vorwiegend graphischem Wege durchgeführt werden kann. Dabei werden ein Routing in Abhängigkeit des Inhaltes (Contentbased-Routing) sowie die Reihenfolgetreue (Sequencing) berücksichtigt. Durch die Business Process Option ist die Abbildung der Integrationsprozesse über mehrere Anwendungen hinweg möglich.

Durch das Produkt Process Integrator ist eine graphische Bearbeitung dieser Integrationsprozesse möglich. Durch die Möglichkeit eines UML-Imports können diese Daten auch aus **CASE**¹¹⁶-Tools bzw. aus Werkzeugen zur Geschäftsprozessmodellierung übernommen werden, sofern diese einen entsprechenden Export unterstützen. Die Administration erfolgt

¹¹⁶ Computer Aided Software Engineering

über Visual Tools und unterstützt auch Scripting. Ein Monitoring der einzelnen Verarbeitungsschritte ist möglich. Es werden verschiedene Logging-Level unterstützt, bei denen auch die übertragenen Daten mitgeloggt werden können. Bei hohen Logging-Leveln können diese Log-Dateien schnell sehr groß werden. Ein spezifisches Logging der Daten für Revisions-Zwecke ist standardmäßig nicht vorgesehen, kann aber einfach über zusätzliche Ausgabeformate realisiert werden. Im Fehlerfall kann eine Signalisierung über Events und/oder e-Mail erfolgen. Es können Statistiken über abgeschlossene Workflows und Performance angezeigt werden. Die e-Link Plattform bietet keine integrierte Versionsverwaltung an, die Versionierung erfolgt in der Regel über Namenskonventionen. Über Import/Export-Funktionen kann eine Versionierung über externe Versionsverwaltungstools erfolgen.

5.2.10 iPlanet: Forté Fusion

Die EAI-Lösung von iPlanet heißt "Forté Fusion" und dies ist auch der Name des Gesamtkonzeptes. Die Lösung ist ein Integrationsserver, der konzeptionell auf XML ausgerichtet ist. Er versucht, durch den XML-Einsatz die individuelle Programmierung auf ein Mindestmaß zu reduzieren. Die Basis von Forté Fusion ist die Forté Entwicklungsumgebung mit Forté **4GL**¹¹⁷, sie integriert sich vollständig in die iPlanet Produktpalette.

Die Transportebene und die Workflow-Steuerung werden von Forté Fusion durch das so genannte „Integration Backbone“ bereitgestellt, einer Teillösung zum Transport und zum Erkennen und Prüfen der Regeln, die für eine Prozesssteuerung erforderlich sind (siehe Abbildung 5.12). Voraussetzung für die Integration von Applikationen ist deren XML-Fähigkeit. Bietet eine Applikation standardmäßig kein XML-Interface an, muss diese Applikation über einen Adapter „XML-enabled“ werden. Darüber hinaus bietet Forté standardmäßig so genannte Technologie-Adapter zu CORBA, TP-Monitoren, Queueing-Systemen usw.. Zur Entwicklung von eigenen Adaptern bietet Forté Fusion ein Adapter Development Kit an.

Die Konnektoren zur Anbindung von Anwendungen basieren auf der XML-Technologie und die Anbindung wird direkt mit XML-Dokumenten realisiert oder es werden - wo dies nicht möglich ist - die Konnektoren zweigeteilt: einerseits anwendungsspezifischer Code zur

¹¹⁷ Fourth Generation Language

Integration in vorhandene Anwendungen, andererseits Code zur Umwandlung der gelesenen Daten in XML-Dokumente. Diese Dokumente beinhalten Business-Objekte wie z. B. Bestellungen, Rechnungen, etc. und werden entsprechend den Regeln der Geschäftsprozesse von einer Anwendung zu einer Anwendung oder mehreren Anwendungen transportiert. Forté Fusion verhält sich wie eine Business Process Engine, die als Laufzeit- und Ausführungsumgebung dient.

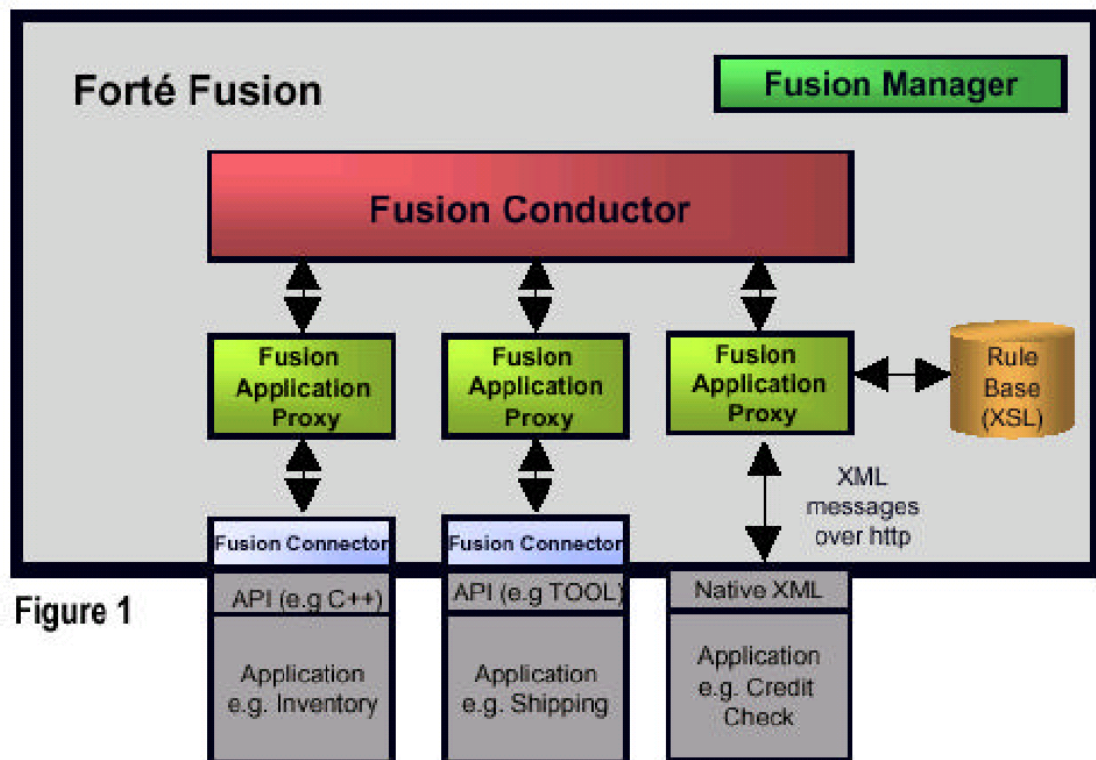


Figure 1

Abbildung 5.12: Forté Fusion Architektur [100]

Als Standard-Transportmechanismus kommen Messaging-Systeme auf Basis JMS und **HTTPS**¹¹⁸ zum Einsatz. Durch Einsatz von JMS kann die Transportschicht durch beliebige Produkte ausgetauscht werden, die JMS entsprechend unterstützen. Standardmäßig arbeitet Forté Fusion mit einem Metadatenmodell, welches konsequenterweise ebenfalls auf XML basiert. Das Datenmapping wird anhand der **XSLT**¹¹⁹ vom Quellformat in das Metaformat und von dort in das Zielformat durchgeführt. Durch dieses Verfahren sind auch n:m Transformationen realisierbar. Alternativ (z. B. aus Performance-Gründen) können die XML-Formate auch direkt ineinander umgewandelt werden, d. h. unter Umgehung des Metaformats. Da Forté Fusion auf der Forté Entwicklungsumgebung basiert, können

¹¹⁸ Hypertext Transfer Protocol (SSL) Secured

¹¹⁹ eXtensible Stylesheet Language Transformations

alternativ zu XML sämtliche Forté-Mechanismen genutzt werden. Der Integrationsprozess kann durch den sogenannten Conductor graphisch bearbeitet werden.

Wie bereits erwähnt, basiert die Standardkommunikation auf HTTPS und JMS. Die Übertragungssicherheit und Verschlüsselung wird somit durch die darunter liegende Transportschicht sichergestellt. Eine Autorisierung wird durch Forté Fusion durch ACL und die Nutzung des Netscape Directory Access, welches auf LDAP basiert, sichergestellt. Hier können gesamte Organisationsstrukturen mit Ihren Rechten entsprechend abgebildet werden. Anwendungen können über ein ADK oder über vorhandene Adapter angebunden werden. Standard-Adapter zu CRM-Systemen wie Clarify, Siebel und Vantive, zu SCM-Systemen wie SAP APO und ERP-Systemen wie Oracle, PeopleSoft und SAP R/3 sind vorhanden, Prebuilt Konnektoren zu Finanzsystemen dagegen nicht. Forté Fusion integriert Applikationen über eine XML-Anbindung. Zur Entwicklung von Adaptern steht ein ADK basierend auf der Forté Entwicklungsumgebung zur Verfügung.

Das Datenmapping erfolgt im Normalfall mit Hilfe der XSLT. Forté Fusion bietet auch hier ein Visual-Tool an, mit dessen Hilfe das Mapping graphisch durchgeführt werden kann. Die wesentlichen Datenbanken können nativ oder über XML angebunden werden. Hinsichtlich Datenformaten werden seitens Forté Fusion XML und Finanzformate wie SWIFT und SAP's IDoc unterstützt. Eine Anbindung von EDI ist über die B2B-Plattform ECXpert möglich. Eine Wandlung von gepacktem EBCDIC-Format erfolgt in der Regel auf Ebene der Adapter. Die vorhandenen Standard-Adapter in Richtung Host-Anbindung unterstützen dies, was bei selbst geschriebenen Adaptern entsprechend zu berücksichtigen ist. Mit dem Forté Fusion Manager bietet iPlanet ein graphisches Tool zum Single-Point-of-Administration. Eine Administration über Scripting wird ebenfalls unterstützt. Ein UMLImport/Export zu CASE-Werkzeugen (z. B. Rational Rose) sowie zum MID Innovator ist vorhanden. Angepasste Oberflächen, z. B. für einen Operator, sind mit Hilfe der Forté Entwicklungsumgebung zu realisieren. Das Regelwerk kann zur Laufzeit aktualisiert werden; laufende Prozesse werden mit dem „alten“ Regelwerk beendet, neue Prozesse greifen auf das neue Regelwerk zu.

Forté Fusion bietet ein Logging auf Prozessebene. Jede Zustandsänderung kann einschließlich der Daten in einer Datenbank gespeichert werden. Über History-Funktionen kann somit ein Audit eines Integrationsprozesses erhalten werden. Abfrageroutinen hinsichtlich der Daten sind für die jeweilige Anforderung eigens zu realisieren. Letzteres gilt auch für statistische Daten, die ebenfalls in einer Datenbank hinterlegt sind. Forté Fusion bietet kein eigenes Versionierungs-Werkzeug an. Da Forté Fusion auf der Forté Entwicklungsumgebung basiert, stehen die dort vorhandenen Mittel vollständig zur Verfügung. Die Forté Entwicklungsumgebung beinhaltet zwar auch kein integriertes

Versionierungswerkzeug, bietet aber Schnittstellen zu 3rd Party Produkten wie PVCS, RCS, usw..

5.2.11 NEON Systems: iWave

NEON Systems ist vor allem durch seine Shadow-Produktreihe bekannt, welche direkten und transparenten Zugriff auf **OS/390**¹²⁰ Systeme erlauben. Mit der iWave Produktreihe bietet NEON Systems ein EAI-Tool an, welches insbesondere durch die Erfahrung von NEON Systems im OS/390 Umfeld eine gute Host Connectivity aufweist.

Die iWave Architektur besteht auf der Transportebene aus dem iWave Director, auf der Konvertierungsebene aus dem iWave Designer, den iWave Interfaces, welche die Connectivity zu den Anwendungen bereitstellen, und dem iWave Administrator, der einen Single-Point-of-Administration bietet. Um eine Anwendung mit iWave zu integrieren, muss auf dem anzubindenden System ein so genannter iWave Client (=Connector) installiert werden (siehe Abbildung 5.13). Dieser Client muss auf der jeweiligen Quell-/Zielplattform gestartet werden.

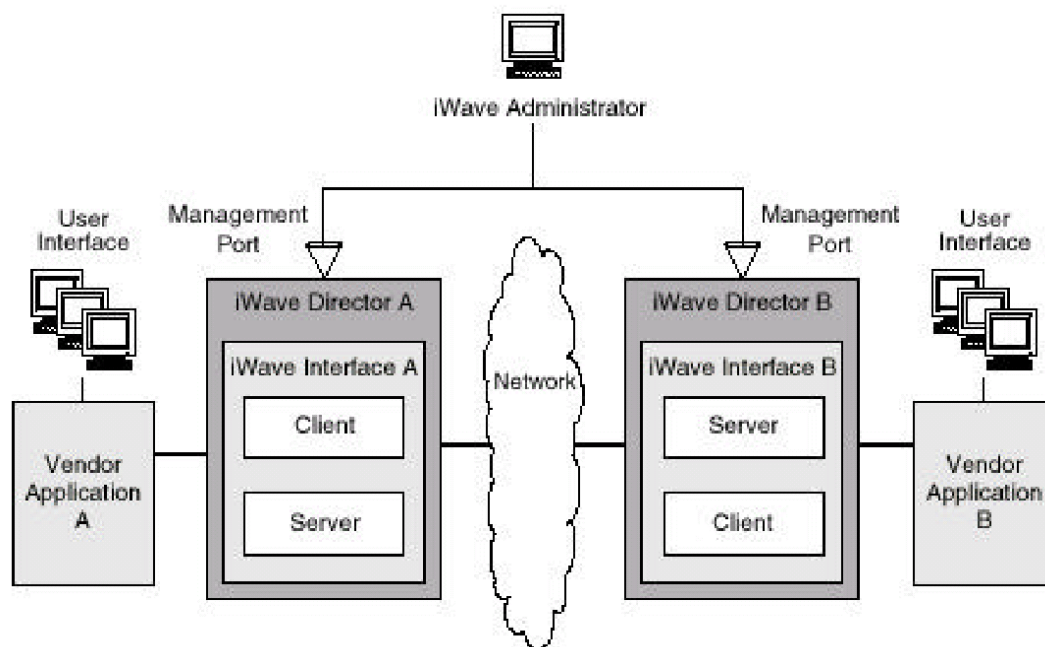


Abbildung 5.13: iWave Architektur [100]

¹²⁰ Operating System /390 (IBM)

Die iWave Clienten arbeiten in einer Client-Server Architektur, d. h. für eine Datenkommunikation wird bei der Zielapplikation ein eigener Serverprozess aufgerufen. Dadurch wird jeweils zwischen zwei miteinander kommunizierenden Applikationen eine Punkt-zu-Punkt Verbindung aufgebaut, was die Anzahl der Schnittstellen nicht verringert, sondern nur einheitliche Mechanismen zur Datenkommunikation zur Verfügung stellt. iWave stellt keine Transaktionssicherheit zur Verfügung und unterstützt demzufolge auch kein 2-Phase-Commit. Ein gegebenenfalls erforderliches Rollback muss manuell realisiert werden. iWave bietet eigene Sicherheitskonzepte und kann auch 3rd Party Tools einbinden. Eine Autorisierung wird sowohl auf User-, Rollen- und Gruppenebene unterstützt, ebenso wie eine LDAP-Anbindung. iWave bietet Standard-Interfaces zu den CRM-Systemen Clarify, Siebel und Vantive sowie zu den ERP-Systemen Oracle Applications und PeopleSoft. Für alle anderen Anwendungen können Interfaces über das Interface Development Kit realisiert werden. Pre-built Interfaces zu Finanzapplikationen sind nicht vorhanden.

Wie weiter oben bereits beschrieben, müssen die so genannten iWave Clienten (=Konnektoren) auf der jeweiligen Quell-/Zielplattform laufen. Ein Client ruft dabei zum Datenaustausch einen Serverprozess auf der Zielapplikation auf. Dadurch wird ein einheitlicher Mechanismus zur Kommunikation verwendet. iWave bietet Native-Zugriff auf alle wesentlichen Datenbanken und unterstützt auch ODBC und JDBC. Allerdings unterstützen diese Standardinterfaces keine Trigger, sondern Pollen die Datenbank, was erhebliche Auswirkungen auf die Performance haben dürfte. Eine Wandlung vom EBCDIC Format erfolgt transparent für den Anwender. Das Datenmapping unterstützt nur 1:1 Mapping, ein n:m Mapping wird derzeit nicht unterstützt. Bei der Unterstützung des Regelwerks gilt zu beachten, dass keine IF/THEN/ELSE Regeln existieren, die ein inhaltsbasiertes Routing erlauben. Die Abbildung des Mappings und des Regelwerks erfolgt script-basiert. Die visuelle Unterstützung ist nur rudimentär.

Die Administration erfolgt ausschließlich script-basiert. Eine Unterstützung durch Visual Tools ist für spätere Releases geplant. Datenlogging wird unterstützt, es können auch die übertragenen Daten geloggt werden. iWave verfügt weder über ein integriertes Versionierungswerkzeug noch über eine Import/Export-Funktionalität. Die Konfigurationsdaten werden in Flat-Files gespeichert, die über ein externes Versionierungstool verwaltet werden können.

5.2.12 Mercator Software: Mercator

Mercator hat eine universelle gleichnamige Datentransformations-Lösung entwickelt. So kann Mercator durch viele andere EAI-Tools als Datentransformationsbaustein eingesetzt werden. Anfang des Jahres 2000 erfolgte die Namensänderung in Mercator Software und zusätzlich wurde der Integration Server der Fa. Novera hinzugefügt. Der Einsatz-Schwerpunkt von Mercator ist der SAP-Markt mit den SAP R/3-Installationen und dem mySAP.com-Marketplace, aber auch SAP R/2 kann in den Integrationsprozess einbezogen werden. Mercator besitzt von SAP zertifizierte Schnittstellen, um SAP R/3 mit Legacy und 3rd Party-Anwendungen zu integrieren. Es besteht eine strategische Allianz mit SWIFT, die Transformation der SWIFT-Meldungen wird daher unterstützt.

Mercator konzentriert sich auf das Internet als Kommunikationskomponente zwischen verschiedenen SAP-Installationen oder auch zur Integration von Legacy-Anwendungen: Das Internet lässt sich wie ein **LAN**¹²¹ nutzen, wenn die Software-Schnittstellen und der Routingmechanismus hinter einer EAI-Infrastruktur versteckt werden können. Für den reinen Datentransport stützt Mercator sich auf 3rd Party Produkte wie MQSeries, BEA Tuxedo, TIB/RV, MSMQ, usw.. Die Konvertierungsebene wird durch den Mercator, die Prozessebene und die externe Integration durch die hinzugekauften und unter dem Namen e-Business Broker vermarktete Suite von Novera abgedeckt. Mercator ist eine C-Applikation, die unter einer Vielzahl verschiedener Plattformen einschließlich Mainframe zur Verfügung steht. Bei keinem der Mitbewerber ist eine so große Zahl unterstützter Plattformen vorhanden.

Mercator „compiliert“ die Transformationsregeln und die Routingregeln in einer Binärdatei, die auf die Zielsysteme verteilt werden müssen. Diese Verteilung kann manuell erfolgen oder mit Hilfe des Integration Flow Designers erfolgen, über den auch die Integrationsprozesse graphisch bearbeitet werden können. Mercator unterstützt ab der Version 5.2 Transaktionen. Durch die Architektur mit MQSeries als Standardtransportsystem müssen die Transaktionen aber in Teiltransaktionen aufgeteilt werden, da mit Einstellen der Daten in das Messaging-System eine Teiltransaktion abgeschlossen ist. Ein Roll-Back muss durch die Applikation unterstützt und im Regelwerk implementiert werden. Für die Gestaltung von Transformationen ist eine Kaskadierung von Transformationsprozessen möglich. Dadurch lässt sich eine Hub-Architektur hinsichtlich der Transformation realisieren. Mercator selbst unterstützt kein Load-Balancing. Dies ist nur in Kombination mit der e-Business-Broker Suite und dem Einsatz eines Application Servers möglich.

¹²¹ Local Area Network

Mercator bietet keine eigenen Verschlüsselungsalgorithmen an, sondern stützt sich auf den zugrundeliegenden Transportmechanismus (z. B. MQSeries). Da die Mercator Transformationsprozesse über das API der Anwendung bzw. per Kommandozeile oder Script in der Execution Engine gestartet werden, erfolgt eine Autorisierung lediglich über das Betriebssystem. Mercator verfügt über keine eigenen Autorisierungsmechanismen oder LDAP-Anbindung, bietet aber Standard-Adapter zu wesentlichen CRM-, SCM- und ERP-Systemen. Insbesondere die Connectivity zu SAP R/3 (und auch zu SAP R/2) ist hervorzuheben. Es existieren auch Konnektoren zu Finanzsystemen wie Kondor+, Murex und Bloomberg.

Mercator bietet eine Entwicklungsumgebung zur Definition des Mappings und Routings ausschließlich auf WindowsNT an. Aus dieser Entwicklungsumgebung wird ein Binärfile erzeugt, das in einer so genannten „Execution Engine“ auf der Zielplattform ausgeführt wird. Der Transformationsprozess wird dabei nur wenn erforderlich gestartet (z. B. über das API einer anderen Anwendung oder über Kommandozeile bzw. Script aufgerufen). Nach erfolgter Transformation und Routing beendet sich der Transformationsprozess wieder. Mercator unterstützt Native-Zugriffe auf die meisten Datenbanken und ODBC. Ein automatisiertes Auslesen vorhandener Datenbankstrukturen zur weiteren Verwendung im Datenmapping wird dabei nicht unterstützt. Mercator unterstützt neben XML, CSV und Flatfile-Anbindung auch diverse Datenformate aus dem Finanzbereich wie Cobol CopyBooks, EDI, FXML, IDoc und SWIFT. Mercator bietet kein eigenes Adapter Development Kit, da diese in C/C++ in jedem beliebigen SDK entwickelt werden können.

Die Administration erfolgt ausschließlich über Visual Tools. Ein spezifisches Logging für Daten kann einfach beim Datenmapping als weitere so genannte „Ausgabekarte“ definiert werden. Mercator bietet keine interne Versionsverwaltung an. Da Mercator alle internen Daten in z. T. binären Flatfiles im Dateisystem hält, können diese Dateien mit einem externen Versionsverwaltungswerkzeug versioniert werden. Neue Regelwerke können zur Laufzeit verteilt werden, die Execution Engine muss aber neu gestartet werden, damit die neue Steuerdatei eingelesen wird. Im Fehlerfall kann eine Signalisierung über Events und/oder e-Mail erfolgen. Ein Account-Management wird nicht unterstützt. Es können Statistiken über abgeschlossene Workflows und Performance angezeigt werden.

5.2.13 SunGard Business Integration: MINT

SunGard Business Integration positioniert sich als Hersteller intelligenter Middleware und EAI-Lösungen im Finanzmarkt. SunGard Business Integration kommt historisch gesehen aus dem SWIFT-Umfeld und hat eine strategische Partnerschaft mit SWIFT. Die MINT Knowledge Family besteht aus MINT Knowledge Broker, MINT Knowledge Manager und dem MINT Publisher (siehe Abbildung 5.14).

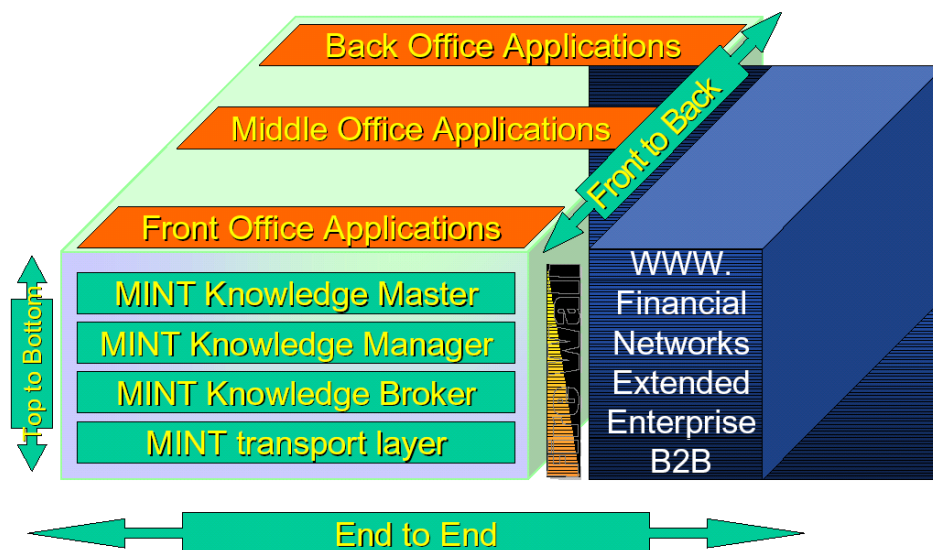


Abbildung 5.14: MINT Knowledge Family Architektur [100]

Die Produkte sind in C++ realisiert und stehen unter Sun Solaris, HP-UX, AIX und WindowsNT zur Verfügung. Transport-, Konvertierungs- und Prozessebene werden durch eigene Produkte abgedeckt. Die Mainframe-Integration erfolgt im Regelfall über die Integration von MQSeries oder BEA Tuxedo, für die entsprechende Adapter bestehen. MINT unterstützt keine Transaktionen oder 2-Phase-Commit. Dies wird der darunter liegenden Transportschicht, z. B. MQSeries überlassen. Ein „Alles-oder-Nichts“ Prinzip muss über Applikationslogik implementiert werden. Es wird kein automatischer Roll-Back unterstützt. Die Reihenfolgetreue bei der Übertragung wird durch das Transportsystem sichergestellt.

MINT bietet Schnittstellen zu externen Security-Produkten wie Security Dynamics ACE Server. Passwörter werden in MINT verschlüsselt gespeichert. MINT bietet eine umfangreiche Autorisierung für User, Rollen und Gruppen. Es kann eine komplette Organisationsform von einer Bank über Bereiche und Abteilungen bis hin zu Einzelpersonen definiert werden. Eine LDAP-Unterstützung besteht derzeit nicht. Anwendungen können über Anbindung der Transportebene (MQSeries, TIBCO, etc.) oder über API-Adapter angebunden

werden. MINT verfügt über wenige Standard-Adapter zu ERP-, SCM- und CRM-Anwendungen. Durch die Spezialisierung auf den Finanzbereich sind einige Adapter zu Finanzanwendungen (z. B. Boss/Cube, Front Arena, Kondor+, SAP) vorhanden. API Adapter müssen in C/C++ realisiert werden und auf der Plattform, auf der die anzubindende Anwendung läuft, installiert werden, ein entsprechender C/C++-Compiler ist hierfür Voraussetzung.

Die Anbindung von Mainframes erfolgt in der Regel über MQSeries. Hinsichtlich des Datenmappings arbeitet MINT nach eigenen Angaben mit einem Metadatenmodell, d. h. n:m Abbildungen sind möglich. Die Regeln werden zur Laufzeit nicht interpretiert, sondern liegen als kompilierter C++ Code vor, der auf der Produktionsmaschine ausgeführt werden muss. Für ein Update des Regelwerks muss ein Integrationsprozess gestoppt, das neue Regelwerk verteilt und der Integrationsprozess neu gestartet werden. Ein Restart des Servers ist dabei nicht erforderlich.

Datenbanken können z. T. nativ oder über ODBC angebunden werden. Ein automatisches Auslesen bestehender Datenbankstrukturen zur Übernahme für das Datenmapping wird nicht unterstützt. Cross-Referencing von Werten werden über Look-Up Tabellen realisiert. Eine Konvertierung von EBCDIC nach **ASCII**¹²² wird nicht automatisch unterstützt, sondern als Funktionalität der darunter liegenden Transportschicht betrachtet. Die Administration erfolgt ausschließlich über Visual Tools. Alle übertragenen Daten können aus Gründen der Revision in einer Datenbank gespeichert werden (Audit-Funktion). Zu Zwecken der Dokumentation können über den MINT Knowledge Publisher diverse Reports und HTML-Exports des Regelwerks erzeugt werden. Im Fehlerfall kann eine Signalisierung über Events und/oder e-Mail erfolgen. Statistiken über Performance oder Änderungen können nicht erzeugt werden. Über das GUI kann der Status von Workflows angezeigt werden. MINT bietet derzeit keine integrierte Versionsverwaltung an. Die Versionierung erfolgt über Namenskonvention.

¹²² American Standard Code of Information Interchange

5.2.14 IBM: MQSeries / MQIntegrator / MQWorkflow

IBM ist im Bereich der Middleware vor allem durch das Produkt MQSeries bekannt. Mit MQSeries ist IBM im Bereich der **MOM**¹²³ Marktführer, nicht zuletzt aufgrund der Verfügbarkeit des Produktes auf den verschiedensten Plattformen. MQSeries wird auch von anderen EAI-Toolherstellern als Transportsystem genutzt oder kann per Adapter bzw. Connector angebunden werden.

Durch den MQ Integrator und MQ Workflow hat IBM eine Ergänzung für MQ Series gefunden, mit der EAI-Lösungen realisiert werden können. MQIntegrator und MQWorkflow sind zwei Produkte, welche die Konvertierungsebene und Prozessebene basierend auf MQSeries als Transportsystem abdecken. MQIntegrator war in der ersten Version eine Gemeinschaftsentwicklung von IBM und NEON. Mit der Version 2 ff setzte bei IBM eine eigene Entwicklung ein, die im Kern allerdings immer noch auf der NEON Engine basiert. Die Integration von Applikationen erfolgt über eine Hub and Spoke Architektur (Abbildung 5.15).

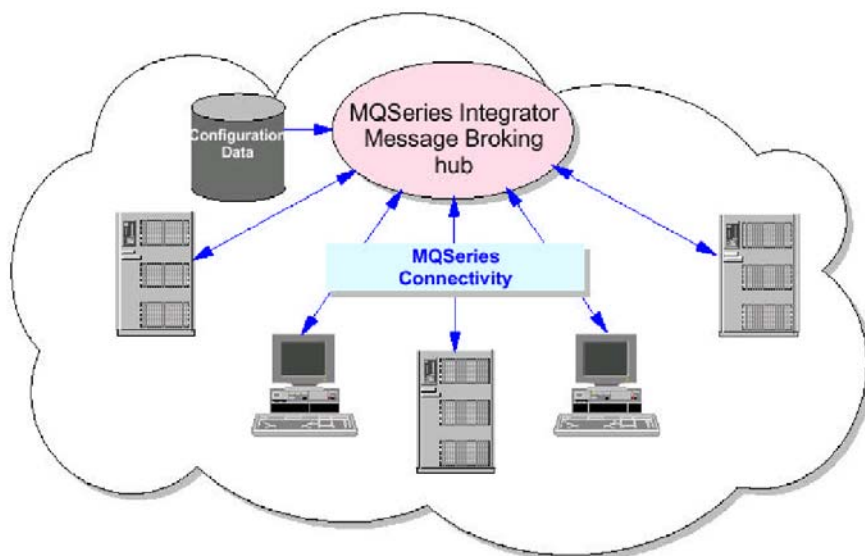


Abbildung 5.15: Hub and Spoke Architektur mit MQIntegrator [100]

Der MQIntegrator arbeitet intern mit Metaobjekten, d. h. Businessobjekte werden für das Mapping in ein internes Format und von dort in das Zielformat abgebildet. Das Routing von Daten kann inhaltsbasiert erfolgen. Aus Performancegründen ist der MQIntegrator eine C++-Applikation. Er steht unter allen wesentlichen Betriebssystemen zur Verfügung. Eine Transaktionssicherheit mit 2-Phase-Commit über mehrere Anwendungen hinweg ist nicht

¹²³ Message Orientated Middleware

möglich, da sich der MQIntegrator als „normale“ MQSeries Anwendung verhält, d. h. wenn aus Sicht von MQSeries eine Transformation/Routing „committed“ wurde, ist für MQSeries die Transaktion abgeschlossen. Ein 2-Phase-Commit über mehrere Anwendungen hinweg ist somit nicht möglich und muss in mehrere Teiltransaktionen aufgeteilt werden. Die Reihenfolgetreue wird durch MQSeries sichergestellt, auch unter dynamischem Load Balancing.

Eine Verschlüsselung auf Transportebene kann über 3rd Party Tools erreicht werden. Hinsichtlich der Autorisierungsmechanismen greifen alle Zugangsmechanismen von MQSeries. MQIntegrator stellt darüber hinaus verschiedene Nutzergruppen zur Verfügung, die in die normalen Zugangsmechanismen des Betriebssystems eingebunden werden können. Eine LDAP-Anbindung ist vorhanden. Die meisten Standardanwendungen bieten Schnittstellen zu MQSeries an. IBM verfolgt keine ausgeprägte Adapter-Politik, sondern verlässt sich bei der Adapter-Entwicklung auf seine Partner, z. B. CrossWorlds, Extricity, NEON, usw.. Die Administration erfolgt im Wesentlichen über Visual Tools. Neue Integrationsprozesse können zur Laufzeit distribuiert und aktiviert werden. Es werden verschiedene Logging-Level unterstützt. Ein Data-Logging kann einfach durch einen weiteren Ausgabepfad bzw. zur Audit-Trails in eine Datenbank realisiert werden.

Die MQSeries Produktfamilie hat keine integrierte Versionsverwaltung. Eine Versionierung kann über Import/Export-Funktionen und Einbindung externer Versionsverwaltungs-Tools erfolgen.

5.2.15 !Candle: ROMA

!Candle positioniert seine CandleNet eBusiness Plattform (vormals ROMA) als Business **SOA**¹²⁴. !Candle nutzt für die Abdeckung der einzelnen Ebenen Standards wie MQSeries, Mercator, StaffWare und stellt ein API zur Verfügung, mit dem die nativen Schnittstellen dieser Standards verdeckt werden. Der Schwerpunkt liegt auf der Bereitstellung von zentralen Services.

Den Ausgangspunkt der CandleNet eBusiness Plattform (im folgenden noch mit ROMA bezeichnet) bildet der Ansatz, dass der Einsatz von Middleware alleine noch kein EAI

¹²⁴ Service Orientated Architecture

bedeutet. Erst die zusätzlichen Funktionen wie Transformation und Routing sowie die Abdeckung komplexer Integrationsprozesse bildet einen Ansatz für EAI. !Candle nutzt diese Dienste als Basis für sein ROMA, welches Standardprodukte zur Abdeckung der Transport-, Konvertierungs- und Prozessebene integriert und zusätzliche Dienste wie Load Balancing, Security Monitoring, Management, usw. ergänzt (siehe Abbildung 5.16).

ROMA nutzt demzufolge keinen eigenen Transportlayer, sondern stützt sich auf MQSeries, BEA MessageQ oder TIB/Rv. Hinsichtlich der Konvertierungsebene bietet ROMA eigene Brokerfunktionalität an, kann aber auch externe Broker wie Mercator und MQIntegrator einbinden. Auf der Prozessebene bietet ROMA Schnittstellen zu StaffWare und MQWorkflow. Ziel von !Candle ist es, Services rund um das Thema Middleware anzubieten, ohne die Middleware direkt programmieren zu müssen. Als Repository verwendet ROMA ein so genanntes Business Service Directory. Dieses basiert auf LDAP und enthält alle Informationen, die zum Betreiben der Business Services benötigt werden.

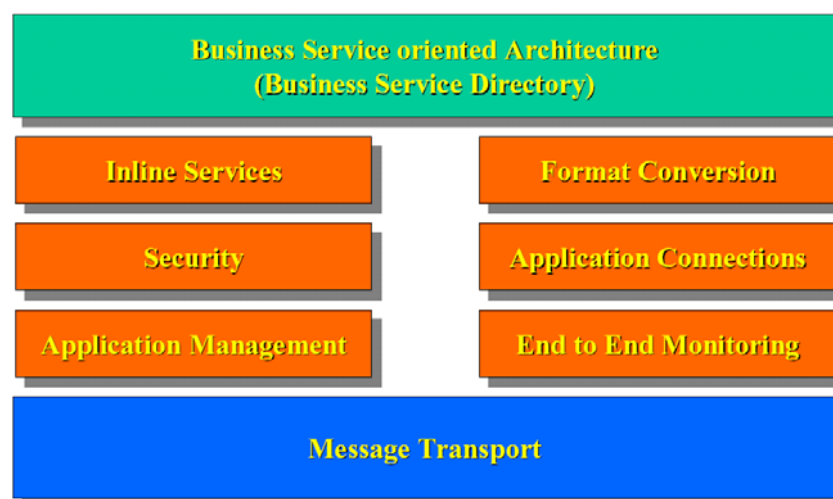


Abbildung 5.16: Architektur CandleNet eBusiness Plattform, vormals ROMA [100]

Transaktionen unter Nutzung eines 2-Phase-Commit über mehrere Anwendungen hinweg werden durch den ROMA Broker nicht unterstützt. In Kombination mit einem TP-Monitor unter Verwendung des eigenen Brokers können solche Transaktionen sichergestellt werden. Wird ein anderer Broker verwendet (Mercator oder MQIntegrator), können 2-Phase-Commit Transaktionen über mehrere Anwendungen hinweg nicht mehr sichergestellt werden. !Candle bietet mit ROMA Secure ein eigenes Security-Produkt an, welches in die CandleNet eBusiness Plattform integriert ist. Damit werden Autorisierung auf User-, Gruppen-, und Rollenebene ermöglicht. Alle Daten werden in einem LDAP-konformen Repository verwaltet.

ROMA bietet nur einige wenige Adapter zu Standardanwendungen an. Dies sind Siebel, SAP R/3 und Peoplesoft, im Finanzbereich wird SWIFT unterstützt. ROMA bietet ein API, das zur Anbindung genutzt werden kann. Hierfür ist die entsprechende Applikation anzupassen bzw. zu "wrappen". Weiterhin bietet ROMA einen so genannten „Universal Connector“. Dieser stellt ein Entwicklungsframework zur Verfügung, über welches Applikationen an ROMA angebunden werden können.

ROMA verfügt derzeit über keine Standard-Adapter, über die nativ auf Datenbanken zugegriffen werden kann. Der Zugriff auf Datenbanken ist über den „Universal Connector“ anwendungsspezifisch zu realisieren. Hinsichtlich der Konvertierung von Daten bietet ROMA eigene Broker-Funktionalitäten an. Das Datenmapping unterstützt dabei den Import/Export von XML (**DTD**¹²⁵) und Cobol Copy-Books. Ein Cross-Referencing von Daten ist über Look-ups möglich. Ein Datenmapping ist nur 1:n, nicht aber n:m möglich. Für komplexeres Datenmapping können Mercator oder MQIntegrator eingebunden werden. Bezüglich des Routings bietet ROMA kein inhaltsbasiertes (content based) Routing an. Dies ist nur durch Einbindung anderer Broker (Mercator, MQIntegrator) möglich. In letzterem Fall ist allerdings keine Transaktionssicherheit über 2-Phase-Commit möglich.

Die Administration erfolgt GUI-basiert über das !Candle Command Center. Ausführliches Monitoring und Audits können ebenfalls über integrierte !Candle Tools durchgeführt werden. Logging-Daten werden in einem Flatfile in einem internen Format gespeichert. Dabei ist auch die Speicherung der übertragenen Daten möglich. Diese Log-Files sind mit !Candle-Tools auswertbar. Gleiches gilt für statistische Daten und Performance Messungen. Eine Versionierung wird in ROMA über Namespaces durchgeführt, wobei ein Deployment des Regelwerkes anhand der Namespaces möglich ist. Ein Versionsverwaltungstool ist nicht integriert. Über Import/Export-Funktionen können 3rd Party Versionierungswerkzeuge wie z. B. PVCS, RCS, usw. genutzt werden.

5.2.16 Compuware: UNIFACE

UNIFACE ist Compuwares Umgebung für die Erstellung, die Aktualisierung und Integration strategischer Anwendungen. Durch die Architektur von UNIFACE eignet sich diese

¹²⁵ Document Type Definition

Entwicklungsumgebung auch zur Enterprise Application Integration von der klassischen A2A bis zur B2B.

UNIFACE ist vorwiegend als Entwicklungsplattform bekannt, EAI wird seitens Compuware als Lösung, nicht als Produkt angeboten. Kern der UNIFACE Entwicklungsumgebung ist eine Komponententechnologie, die über einen eigenen ORB (nicht IIOP Standard) verbunden ist. Compuware steht auf allen relevanten Plattformen zur Verfügung, insbesondere durch eine starke OS/390 Unterstützung ist ein gute Host-Connectivity gegeben. UNIFACE unterstützt von Haus aus keine Transaktionen, diese müssen programmiert werden, und kann nicht als 2-Phase-Commit Coordinator fungieren.

In UNIFACE können zur Kryptierung der zu übertragenen Daten 3rd Party Produkte eingebunden werden. Die UNIFACE Entwicklungsumgebung beinhaltet eigene Autorisierungsmechanismen und Zugangskontrollen auf User-, Rollen- und Gruppenebene, ebenfalls ist eine LDAP-Unterstützung vorhanden. Bei der Connectivity zu Standardanwendungen fällt negativ auf, dass es sich bei UNIFACE im Wesentlichen um eine Entwicklungsumgebung handelt. So gibt es außer zu SAP R/3 keine Adapter zu Standardapplikationen. Ebenso sind keine Standardadapter für Finanzanwendungen vorhanden. Da es sich bei UNIFACE um eine eigene Entwicklungsumgebung handelt, ist kein separates ADK zur Adapterentwicklung erforderlich, sondern dies kann direkt in der UNIFACE Oberfläche erfolgen. UNIFACE hat einen eigenen Application Server, kann aber auch BEA WebLogic, IBM WebSphere und den Inprise Application Server anbinden.

Neben der eigenen CORBA Implementierung, die leider nicht dem IIOP-Standard entspricht, können auch native Kommunikationsprotokolle wie RPC und RMI sowie MOM (z. B. MQSeries, MSMQ oder Oracle Queue) als Transportsystem genutzt werden. Der Native-Zugriff auf alle relevanten Datenbanken sowie ODBC und JDBC werden entsprechend unterstützt. Die Administration erfolgt über Visual Tools, kann aber auch kommandozeilenbasiert erfolgen. UNIFACE unterstützt als Entwicklungsumgebung voll den UML Standard und verfügt über mächtige Debugging und Tracing-Funktionen. Ein Logging der Daten muss jedoch eigens programmiert werden. Die Versionsverwaltung erfolgt bei UNIFACE über 3rd Party Tools (z. B. PVCS, Continuous), die für den Anwender aber völlig transparent in die Entwicklungsumgebung eingebunden sind.

5.2.17 WebMethods: WebMethods

WebMethods positioniert sich als Unternehmen, welches die gesamte Integrationspalette von Inter-Enterprise Integration über Mainframe-Integration bis zur B2B-Integration abdeckt. WebMethods hat sich speziell auf Großkunden spezialisiert, was sich auch im Lizenzmodell widerspiegelt.

Alle WebMethods Enterprise Komponenten verfügen über ein GUI, welches dafür ausgelegt ist, auch Mitarbeitern, die nicht mit der Programmierung vertraut sind, eine bestmögliche Unterstützung zu geben. WebMethods Enterprise (vormals ActiveWorks Integration System) ist ein Message Broker, der über eine eigene, auf TCP/IP basierende, Transportschicht eine netzwerkverteilte Architektur und damit Skalierbarkeit, Fail-Over und Load-Balancing ermöglicht (vgl. Abbildung 5.17).

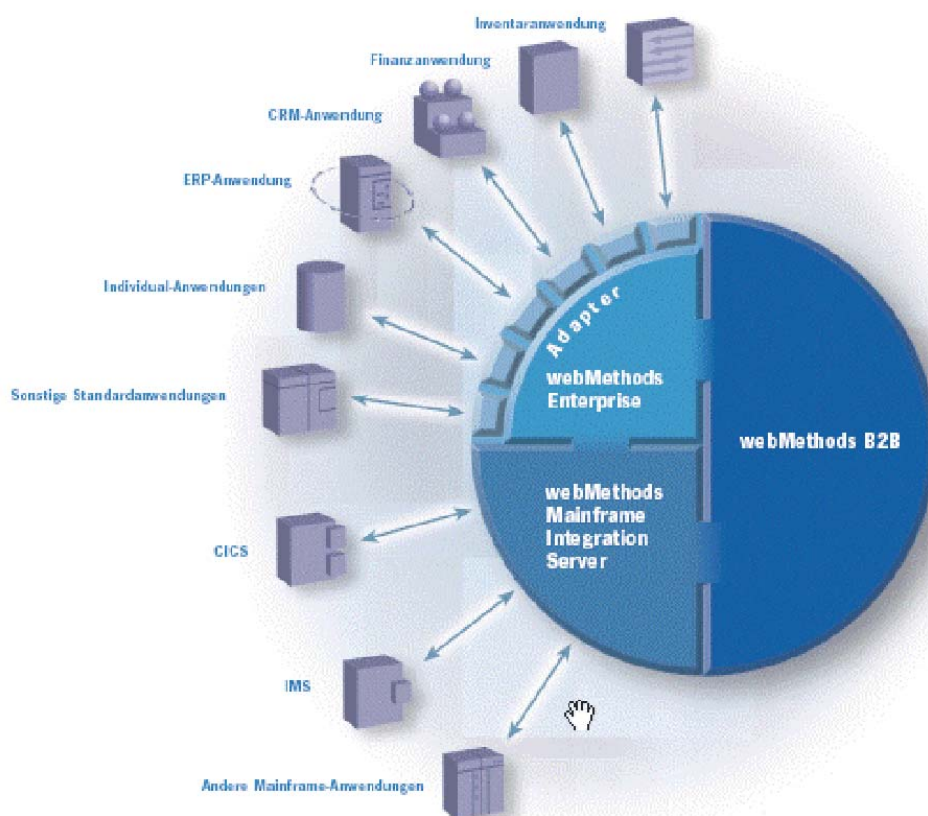


Abbildung 5.17: WebMethods Enterprise Architektur [100]

Über den WebMethods Mainframe Integration Server bietet WebMethods die Möglichkeit, Host- und Mainframe-Applikationen zu integrieren. Konvertierungs- und Prozessebene werden durch eigene Produkte abgedeckt. WebMethods Enterprise steht unter AIX, HP-UX, WindowsNT und Sun Solaris zur Verfügung. Über den WebMethods Mainframe Integration

Server wird ebenfalls OS/390 unterstützt. Adapter stehen unter allen wesentlichen Plattformen zur Verfügung. Sollte eine Plattform nicht direkt unterstützt werden, besteht i. A. die Möglichkeit, die anzubindende Applikation remote anzubinden (z. B. via Distributed API, Dateischnittstelle, etc.). Dies bedeutet auch, dass der Adapter nicht auf der Zielplattform laufen muss. Der Kernel von WebMethods ist aus Performancegründen in C++ realisiert. Die weiteren Programmteile sind als Java Applikationen (compiliert) realisiert.

WebMethods Enterprise bietet einen **ATC**¹²⁶, mit dem langlebige Transaktionen über Anwendungen hinweg durchgeführt werden können. Ein „Alles oder Nichts“ sowie „Guaranteed Delivery“ wird unterstützt, jedoch kein 2-Phase-Commit über mehrere Anwendungen hinweg. Logische Transaktionen über mehrere Anwendungen hinweg müssen zum Teil programmiert werden. Die Reihenfolgetreue ist auch im Fall eines Load-Balancing gewährleistet. Im Falle eines Fehlers einer Applikation kann ein kompletter Roll-Back gefahren werden. Da WebMethods mit verschiedenen Brokern im Netzwerk verteilt arbeiten kann, wird auch eine Fail-Over-Strategie unterstützt, d. h. im Falle des Ausfalls eines Knotens (Brokers) kann ein anderer Broker dessen Aufgabe übernehmen und die Daten entsprechend routen.

WebMethods beinhaltet eine interne Verschlüsselung und kann auch auf Standards wie SSL zurückgreifen. Eine Autorisierung erfolgt über die Mittel des Betriebssystems. Eine spezielle Autorisierung innerhalb WebMethods Enterprise ist nicht vorgesehen. Derzeit ist eine Vielzahl vorgefertigter Adapter verfügbar. Zusätzlich sind ca. 70 Adapter im Rahmen von Kundenprojekten entstanden, die zwar nicht auf der offiziellen Preisliste stehen, bei Bedarf aber angepasst werden können. WebMethods bietet so genannte Application Adapter für die Integration von Standardanwendungen an. Darüber hinaus bestehen Datenbank-Adapter, Language-Adapter (z. B. CORBA, Java, XML) und Mainframe-Adapter (z. B. CICS/COBOL, IMS). Zur Adapterentwicklung steht ein eigenes ADK zur Verfügung, über welches Adapter mit eigenem GUI realisiert werden können.

WebMethods und IDS Scheer planen, eine strategische Partnerschaft einzugehen. Dabei ist eine Übernahme von Geschäftsprozessmodellen aus ARIS in WebMethods Enterprise geplant. Die Administration der verteilten Broker-Architektur kann zentral von einem Arbeitsplatz erfolgen (Single-Point-of-Administration). Zur Administration stehen mehrere Visual-Tools zur Verfügung, die jeweils Spezialaufgaben haben (Enterprise Manager, Adapter Configuration, Adapter Errors Tool, Event Type Editor, Event Monitor). Zusätzlich

¹²⁶ Application Transaction Coordinator

wird ein Scripting unterstützt. Das Regelwerk wird standardmäßig in einem Flatfile gespeichert, kann aber auch in einer Datenbank hinterlegt werden. WebMethods unterstützt derzeit keine integrierte Versionskontrolle. Die Versionierung ist derzeit über Namenskonventionen vorgesehen. WebMethods Enterprise verfügt über Import/Export-Funktionen, wodurch externe Versionsverwaltungssysteme eingesetzt werden können. WebMethods arbeitet im Regelfall mit Partnern (**SI**¹²⁷) zusammen, die das Spezialwissen der entsprechenden Branche mitbringen, wobei WebMethods sich auf wenige, aber wichtige Partner spezialisiert.

WebMethods bietet ein interessantes Lizenzmodell an: Die Lizenzen müssen alle 2 Jahre erneuert werden, d. h. alle 2 Jahre neu bezahlt werden. Dafür erhält der Anwender den vollen Support (24*7 muss separat beauftragt werden) bei entsprechend niedrigen Raten. Für eine Entwicklungslizenz fallen ca. 150.000,00 US\$ an, hierbei gibt es aber keine Beschränkung hinsichtlich Anzahl eingesetzter Installationen oder User. Sie kann im Rahmen dieser Gebühr frei genutzt werden. Ebenso ist ein Einsatz unter verschiedenen Plattformen und Nutzung aller vorhandenen Standard-Adapter möglich. Für die Produktionslizenz fallen Kosten in Höhe von 100.000,00 US\$ je CPU an. Die HA-Option muss separat beauftragt werden.

5.3 Beurteilung der EAI-Systeme zwecks Eignung für das Konzept

Die zuvor genannten EAI-Lösungen stellen zum Großteil die Basis für eine vollintegrierte Systemstruktur dar. Sie sind funktional leistungsfähig und decken technologisch betrachtet in weiten Teilen die Anforderungen an ein klassisches EAI-System aus Kapitel 3 ab.

Sie besitzen andererseits jedoch Eigenschaften, die sie als universelles Integrationsmedium ungeeignet erscheinen lassen, v. a. für die mittelständische Automobilindustrie. Insbesondere gehen Komplexität und Leistungsfähigkeit der vorgestellten EAI-Lösungen zumeist weit über den Bedarf von mittelständischen Unternehmen hinaus. Dies spiegelt sich auch in einem hohen Administrationsaufwand wider, der für den produktiven Systemeinsatz aufzubringen ist. Aus Integrationssicht liegen die Anwendungsschwerpunkte im Finanzbereich sowie im SAP-Umfeld, wodurch die Ausrichtung der Systeme auf Großunternehmen unterstrichen wird. Die Skalierbarkeit ist nur im Hinblick auf Integrationsfunktionalitäten gegeben, d. h. auch wenn durch eine bestimmte

¹²⁷ Systemintegratoren

Integrationskomponente letztlich nur ein einziger Prozess unterstützt wird, so ist dennoch die Installation der gesamten für diese Integrationsaufgabe erforderlichen Module notwendig.

Die betrachteten Systeme besitzen darüber hinaus kaum eine Ausrichtung auf SCM-Konzepte bzw. Automotive-spezifische Anforderungen und bestenfalls eine rudimentäre Applikationsserver-Komponente, die zu einem hohen Ressourcenbedarf führt [105, 106, 107]. Dies führt letztlich zu deren Hauptproblem, das in den zugrunde liegenden Preis- und Kostenstrukturen zu suchen ist. Eine zentrale Bedeutung besitzt hierbei die Investitionshürde, die durch Lizenz-, Implementierungs- und Wartungsaufwand aufgebaut wird, wodurch schließlich der wirtschaftliche Einsatz der betrachteten EAI-Lösungen in der mittelständischen Automobilindustrie in Frage gestellt wird.

Zum besseren Verständnis bezüglich der Eignung der hier analysierten EAI-Systeme für das Konzept sind die wichtigsten Kriterien aus den Anforderungen gemäß Kapitel 3 in der nachfolgenden Tabelle 3 einer Bewertung unterzogen. Dabei entsprechen die in dieser Tabelle genannten Kriterien (K1 – K10) den Anforderungen folgendermaßen:

Anforderungen aus Unternehmensicht (vgl. Kapitel 3.1)

1. Kosten (in Tsd. EURO) für Lizenz- und Implementierungsaufwand **[K1]**
2. Unterstützung von SCM-Konzepten **[K2]**
3. Berücksichtigung der schrittweise wachsenden Integrationsanforderungen / Skalierbarkeit **[K3]**

Anforderung bezüglich der Kommunikation (vgl. Kapitel 3.2)

1. A2A-Beziehung
2. B2B-Beziehung **[K4]**
3. interne und externe Integration / SCM-Prozesse **[K5]**

Anforderung an die Schnittstellen (vgl. Kapitel 3.3)

1. Zentralisiertes Schnittstellensystem **[K6]**

Anforderungen im Zusammenhang mit der Integration (vgl. Kapitel 3.4)

1. Transport von Nachrichten in Echtzeit **[K7]**
2. Unterschiedliche Kommunikationsstile
3. Genormte Nachrichtenformate
4. Datentransformation und -konvertierung
5. Unterstützung von Regeln und Prozessen
6. Konnektivität
7. effiziente Ressourcennutzung **[K8]**
8. Sicherheitskonzepte
9. Scheduling von Nachrichten und Routing
10. Load Balancing, Failover, Monitoring, Recovery-Fähigkeit, Tracing und Debugging

EAI-System	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
1. Alliance Manager	0	1	1	3	1	1	2	0	2	41%
2. CrossWorlds	0	1	2	3	1	0	1	0	2	37%
3. Forté Fusion	0	1	2	2	1	0	2	0	2	37%
4. UNIFACE	0	0	0	1	0	2	2	0	2	26%
5. Active Enterprise	0	0	1	1	0	0	2	0	2	22%
6. Business Ware	0	0	1	1	0	0	1	0	2	19%
7. e*Gate / e*Xchange	0	1	2	3	1	0	2	0	2	41%
8. IWave	0	0	0	1	0	1	1	0	2	19%
9. Mercator	0	1	1	2	1	0	2	0	3	37%
10. MINT	0	0	0	1	0	0	2	0	2	19%
11. ROMA	0	0	1	1	0	0	1	0	1	15%
12. WebMethods	0	1	0	2	1	2	3	0	2	41%
13. Constellar Hub	0	1	1	2	1	1	2	0	3	41%
14. e-Biz Integrator	0	1	2	2	1	0	2	0	2	37%
15. eLink/Tuxedo/WebLogic	0	1	1	2	1	0	2	0	2	33%
16. MQSeries/MQIntegrator/ MQWorkflow	0	0	0	1	0	2	2	0	2	26%
17. BizTalkServer 2002	1	1	1	1	1	0	2	0	2	33%

Tabelle 3: Bewertung der EAI-Systeme

Anforderungen an die Integrationsmethoden (vgl. Kapitel 3.4)

1. Integration über Benutzerschnittstelle
2. Integration über Funktionsaufrufe
3. Integration über Datenbanken und Dateistrukturen **[K9]**
4. Integration über Komponenten

Für die hier vorgenommene Bewertung gilt:

- 0** = nicht geeignet
- 1** = wenig geeignet
- 2** = gut geeignet
- 3** = sehr gut geeignet

Das Kriterium **K10** definiert in Form einer Prozentzahl die Eignung des jeweiligen EAI-Systems für das hier angestrebte Konzept einer Anwendungsintegration, wobei die Einzelbewertungen addiert und zur maximal möglichen Punktzahl in Beziehung gebracht werden. An den niedrigen Prozentzahlen (die größte Übereinstimmung mit den Anforderungen liegt bei 41%, die geringste bei 15%) lässt sich ablesen, dass es an mittelstandstauglichen EAI-Lösungen fehlt, die die betrachteten Kriterien ausreichend erfüllen.

Für einen Lösungsansatz, der den Forderungen dennoch gerecht wird und einen einfachen Einstieg in das Thema EAI mit schnell anwendbaren Konzepten bzgl. des Supply Chain Managements ermöglicht, wird daher im folgenden ein Ansatz gewählt und auch umgesetzt, der eine IT-Umgebung in den Unternehmen nutzt, die häufig bereits vorhanden ist (vgl. Kapitel 4.3). Wenn dieser Lösungsansatz im Ergebnis sicherlich nicht zu 100% vergleichbar ist mit den Möglichkeiten einzelner EAI-Systeme, so ist zumindest sichergestellt, dass die hier formulierte Lösung die mittelständischen Unternehmen in die Lage versetzt, im Hinblick auf ihre internen SCM-Prozesse die externen SCM-Prozessen von Kunde und Lieferant zu integrieren.

5.4 LAMP als Grundlage für die Anwendungsintegration

Heutige EAI-Lösungen sind komplex, teuer und aufwendig zu implementieren. Im unternehmerischen Mittelstand werden jedoch sowohl von der Kosten-, als auch von der

Prozessseite her flexible Leistungsansätze benötigt. Deshalb sind neue Lösungskonzepte erforderlich, um ein globales Geschäftsprozessmanagement in den mittelständischen Supply Chains umsetzen zu können. Demgegenüber existieren bereits heute einzelne Geschäftsprozesse, bei denen das Internet das zentrale Kommunikationsmedium darstellt. Es ist offensichtlich, dass Internettechnologien auch in Zukunft maßgeblichen Einfluß auf Prozessverbesserungen haben werden. Abgeleitet von den Zielen des **WBEM**¹²⁸ [108] stehen Schlüsselthemen wie e-business on demand oder Webservices an der Grenze, sich zum Mainstream zu entwickeln. Die Vorteile von Webservices sind nicht mehr hypothetisch, wie es vor einigen Jahren im Internet-Hype noch der Fall war, sondern für jeden greifbar geworden. Der Zugriff auf Informationen von Geschäftspartnern wird ermöglicht, Mitarbeiter, Partner und Kunden werden in die Lage versetzt, vollständig an Prozessen zu partizipieren und es wird einfacher, Geschäftspartner zu wechseln und hinzuzunehmen. Parallel dazu sinken auf IT-Ebene die Connectivity-Kosten und es wird eine Unabhängigkeit von Plattformen und Techniken geschaffen.

Da in den heutigen Unternehmen die SCM-Prozesse mit zahlreichen unterschiedlichen Anwendungssystemen abgebildet werden, sind EAI-Systeme auf der einen Seite und Webservices auf der anderen alleine nicht ausreichend, um eine zeitgemäße Prozessintegration sicherzustellen. Daher liegt es nahe, im Rahmen eines Lösungskonzeptes die Bereiche Webservices und Anwendungsintegration miteinander zu verbinden, woraus schließlich ein webbasiertes Anwendungsintegrationssystem entsteht. Durch den webbasierten Integrationsansatz wird es möglich, die Vorteile von Webservices und Anwendungsintegration zu vereinen, also einerseits unter Nutzung der Internettechnologien die Abbildung der Prozesse transparent für die Prozessbeteiligten realtime zu gestalten und andererseits bestehende IT-Assets weiter- und wiederzuverwenden bzw. einzubinden. In besonderem Maße geeignet für die webbasierte Anwendungsintegration ist die so genannte LAMP-Umgebung, da sie einerseits im Bereich der Webservices weit verbreitet ist und andererseits eine Reihe von Anforderungen an eine Integrationsplattform abdeckt. Sie beseitigt darüber hinaus die beschriebenen und für den unternehmerischen Mittelstand bedeutsamen Schwächen der existierenden EAI-Systeme. Mit Blickrichtung auf die SCM-Prozesse des automobilen Mittelstandes stellt daher die webbasierte Anwendungsintegration mit LAMP die Basis des auf den folgenden Seiten dargestellten Lösungskonzeptes dar.

Bei dem Ansatz der webbasierten Anwendungsintegration mit LAMP wird das Prinzip verfolgt, die unternehmensseitig vorhandene Standard- und Individualsoftware mit individuell programmierbaren Modulen zu ergänzen, um damit dynamische Webseiten zu erzeugen, die

¹²⁸ Web-Based Management

ihre Inhalte aus verschiedensten Anwendungssystemen beziehen, die serverseitig aufbereitet und über einen Webbrowser ausgegeben werden. Typische SCM-Prozesse werden damit unter Berücksichtigung eines geeigneten Leistungs- und Kostengefüges bedarfsgerecht implementiert. Da LAMP-Strukturen bereits in vielen Unternehmen als Internet- oder Intranet-Server im Einsatz sind, entsteht durch die funktionale Ausweitung zum EAI-Server grundsätzlich kein zusätzliches System. Vielmehr wird hier eine vorhandene LAMP-Installation durch gezielte Erweiterungen EAI-fähig gemacht. Diese Erweiterungen werden in Form von Programmskripten realisiert, da mit LAMP neben einem integrierenden Anwendungssystem auch direkt eine Programmierumgebung zur Verfügung steht.

Technologisch betrachtet besteht LAMP aus den Software-Teilsystemen

Linux, Apache, MySQL und PHP,

woher auch das daraus gebildete Akronym seinen Namen erhält. Durch jedes dieser Einzelsysteme werden spezifische Programmfunktionalitäten abgebildet, die in der Summe ein leistungsfähiges Integrationssystem ergeben.

LAMP wird allgemein als Basissystem für Webservices zur Generierung dynamischer Webseiten eingesetzt. Alle vier LAMP-Komponenten zählen zu Open Source Software, werden unabhängig voneinander unter Bekanntmachung des Quellcodes weiterentwickelt und weisen keine Lizenzgebühren auf. Die erste LAMP-Komponente stellt das Betriebssystem Linux dar, das auf einem Standard IBM-PC installiert werden kann. Es handelt sich um eine zuverlässige und weit verbreitete Anwendungsplattform, die hauptsächlich für Serverdienste und in diesem Zusammenhang für unterschiedliche Zwecke wie File- oder Webservices eingesetzt wird. Neben der kostenfreien Einsatzmöglichkeit hat sich Linux durch die hohe Verfügbarkeit seiner Serverdienste etabliert, im Vergleich zu anderen Serverbetriebssystemen wie **SUN**¹²⁹ Solaris und Microsoft Windows ist Linux daher nicht nur in den Lizenz-, sondern auch in den operativen Kosten deutlich günstiger als seine Mitbewerber (Abbildung 5.18) [109, 110], was letztlich den Erfolg des Betriebssystems begründet.

¹²⁹ Stanford University Network

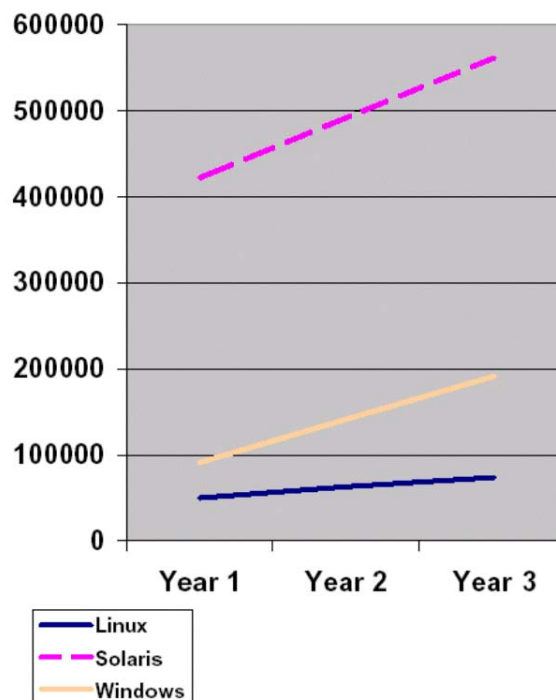


Abbildung 5.18: Total Cost of Ownership bei Linux, Solaris und Windows in US \$ [109]

Den zweiten LAMP-Bestandteil bildet der für Webservices weltweit am häufigsten eingesetzte Webserver Apache. Sein verbreiteter Einsatz resultiert dabei im Wesentlichen aus der Verfügbarkeit für eine Vielzahl von unterschiedlichen Betriebssystemen (AIX, Digital UNIX, FreeBSD, Irix, Mac OS, Netware, OpenBSD, Solaris, SunOS, Windows, Linux). Neben der Einsatzmöglichkeit als Applikationsserver schafft der Apache Webserver die Voraussetzung für die Verarbeitung von Webseiten und damit für eine webbasierte Systemintegration, er liefert die aufbereiteten Informationen plattformunabhängig an jeden beliebigen Webbrowser, so dass praktisch keine systemimmanenten Nutzungsgrenzen existieren [111, 112, 113].

Durch die Verwendung des Datenbanksystems MySQL, das die dritte LAMP-Komponente darstellt, steht die Infrastruktur für ein Datawarehouse zur Verfügung, wie sie für EAI-Systeme benötigt wird. MySQL besitzt insbesondere im Vergleich zu Datenbanksystemen von Oracle oder Microsoft Performancevorteile und bietet erweiterte Funktionen für Skriptsprachen wie PHP.

PHP dient zur Erstellung bzw. Generierung dynamischer Webseiten, die über den Webserver ausgegeben werden und stellt zugleich die vierte Komponente der LAMP-Plattform dar. Dynamische Webseiten werden im Gegensatz zu statischen Seiten zur Laufzeit, also im Moment des Anforderns, erzeugt und enthalten demzufolge variable Informationen, beispielsweise aus Datenbanken oder anderen Datenquellen. PHP ist dabei

keine eigenständige Anwendung im klassischen Sinn, sondern wird als Modul in den Apache Webserver eingebunden. PHP-Skripte können daher direkt in HTML-Code integriert werden und müssen hierzu einerseits im HTML-Code durch Start- und Endtags (<? oder <?php bzw. ?>) gekennzeichnet und andererseits mit der Dateierweiterung .php versehen werden. Die Verarbeitung des PHP-Codes erfolgt über den PHP-Parser, der durch die Dateierweiterung .php aufgerufen wird und den durch Semikola zeilenweise getrennten PHP-Code interpretiert. Zur Veranschaulichung der PHP-Syntax ist in der Abbildung 5.19 die PHP-Datei hello.php dargestellt, über die letztlich der Text „Hello World“ im HTML-Format über den Webbrowser ausgegeben wird.

```
<html>
<head> <title>PHP Test</title></head>
<body>

<?
echo "<h1>Hello World";
?>

</body>
</html>
```

Abbildung 5.19: PHP-Syntax am Beispiel des Skriptes hello.php

Der Dateiaufbau entspricht einer typischen HTML-Syntax, in der der PHP-Code durch die PHP-Tags eingefügt ist.

Von PHP werden mittlerweile weit über 100 Funktionen unterstützt, darunter mathematische Funktionen, Komprimierungsalgorithmen (z. B. Bzip2), Netz- und Verzeichnisdienste (z. B. Email, ftp, LDAP), Netzprotokolle (z. B. XML, **SOAP**¹³⁰) oder Anwendungsfunktionen (z. B. COM, .NET). Besondere Bedeutung besitzt die sehr umfangreiche native Datenbank- (z. B. MySQL, dBase, Informix) und Datenbanksystemunterstützung (z. B. Oracle 8, Lotus Notes) von PHP, die einen großen Vorteil gegenüber den konkurrierenden Varianten zur Erzeugung dynamischer Inhalte über serverseitigen Skriptsprachen (**CGI**¹³¹, ASP, Cold Fusion, Servlets, **JSP**¹³²) darstellt und dadurch ein breites Anwendungsspektrum für die Anwendungsintegration bietet. Bei nicht nativ unterstützten Datenbankformaten (z. B. DB2, Adabas D) stehen für PHP eine Vielzahl von ODBC-Konnektoren zur Verfügung, so dass es kaum Grenzen gibt, um eine Datenbank-Connectivity zu Datenbanken unterschiedlichster

¹³⁰ Simple Object Access Protocol

¹³¹ Common Gateway Interface

¹³² Java Server Pages

Hersteller mit PHP herzustellen. Auf diese Weise fungiert PHP als Multiadapter für die integrierenden internen und externen Datenquellen, Systeme und Anwendungen, auch im Hinblick einer standardisierten Kommunikation zwischen internen und externen Systemen mittels XML.

Für die Anbindung von Datenbanken mit PHP sind nur wenige Zeilen PHP-Code erforderlich. Um beispielsweise eine Verbindung zu einer MySQL-Datenbank herzustellen, die auf dem LAMP- oder einem entfernten System installiert ist, wird neben den Anmeldeinformationen für die Datenbank selbst nur der für MySQL-Datenbanken verfügbare PHP-Funktionsaufruf `mysql_connect` benötigt (vgl. Abbildung 5.20). Hier stellen die Variablen `$server` die IP-Adresse des Systems dar, auf dem sich der MySQL-Server befindet, und `$user` sowie `$pass` die Anmeldedaten für die Datenbank `$dbase`. Die in PHP mit zwei Schrägstrichen angegebenen Kommentare dienen dabei lediglich der Beschreibung, ob die Anmeldung erfolgreich war oder fehlgeschlagen ist.

Neben Datenbankstrukturen lassen sich auf Fileebene über zahlreiche Dateisystemfunktionen auch andere Datenquellen über PHP ansprechen (z. B. Dateien, Texte). Flatfiles oder Legacy¹³³-Formate können eingebunden werden, zusätzlich erfolgt durch die kontinuierliche Weiterentwicklung auf Open Source-Ebene eine stetige quantitative und funktionelle Ausweitung der PHP-Funktionen.

```
<?
$server="192.168.1.1";
$user="test";
$pass="hello";
$dbase="datenbank1";
$conn=@mysql_connect($server,$user,$pass);
if($conn) { mysql_select_db($dbase, $conn);
            // Datenbankanzbindung erfolgt
        } else { die;
                // Datenbankanzbindung nicht erfolgt
            }
?>
```

Abbildung 5.20: MySQL-Datenbankanbindung mit PHP

In Verbindung mit dem Datenbanksystem MySQL stellt PHP die Grundlage für den Aufbau eines Data Warehouses dar. Im Hinblick auf hierbei erforderliche Transformationen der

¹³³ als Legacy-Systeme werden Anwendung bzw. Softwaresysteme bezeichnet, die unstrukturiert, und schlecht überschaubar sind und damit nur noch unter hohen Kosten erweitert werden können

Datenformate, die im Rahmen von Integrationsanforderungen benötigt werden, unterstützt PHP den Aufbau zentraler Repositories, indem die benötigten Datenformate skriptgesteuert eingelesen und dem Regelwerk des Repositories entsprechend normalisiert ausgegeben werden, beispielsweise direkt als Webseite oder in eine MySQL-Datenbank.

Um nun die Eignung von LAMP als webbasierte EAI-Plattform einordnen zu können, werden im Folgenden die Komponenten detaillierter betrachtet, die für eine webbasierte Anwendungsintegration erforderlich sind. Dazu wird zunächst von einem allgemeinen Aufbau einer webbasierten Integrationsplattform ausgegangen. Hier sind vier wesentliche Elemente vorhanden, nämlich

- Legacy-Systeme, die zu integrieren sind, wie CAD-, ERP-, BDE-System
- die EAI-Komponente
- eine Kommunikationsinfrastruktur, über die alle Systeme miteinander verbunden werden
- Webservices-Frontends, die die benötigten Informationen über Webtechnologien zur Verfügung stellen.

Während die Legacy-Systeme die zu integrierenden System darstellen, kommen der EAI-Komponente sowie der Kommunikationsinfrastruktur gemeinsam die Integrationsaufgaben zu, hier sind die Schnittstellen zu den Legacy-Systemen erforderlich. Genau an dieser Stelle wird die LAMP-Umgebung eingesetzt, indem das System durch Daten- und Datenbankverbindungen auf Basis der Kommunikationsprotokolle TCP/IP zum zentralen Integrationsserver gestaltet wird (vgl. Abbildung 5.21). Dabei werden auf der Seite des Webservice-Frontends (Webbrowser) die Anfragen an den LAMP-Server zunächst als in HTML eingebetteter PHP-Code weitergeleitet. Hier übergibt der Apache Webserver den enthaltenen PHP-Code an den PHP-Parser, der die Verbindungen zu Datenbanken in Form von SQL-Syntax und zu den Datenstrukturen durch PHP-spezifische Funktionen herstellt. Mit PHP-Code werden die spezifischen Informationen aus den einzelnen Datenbank Anwendungen bzw. Dateistrukturen verknüpft, ggf. über ein definiertes XML-Schema bzw. Repository aufbereitet und entweder in einer MySQL-Datenbank zusammengeführt oder direkt über den Webserver als HTML- oder XML-Code an den Webbrowser ausgegeben. Alternativ dazu ist der Versand der Daten als Email möglich oder via ftp, http, usw. als Datenstream.

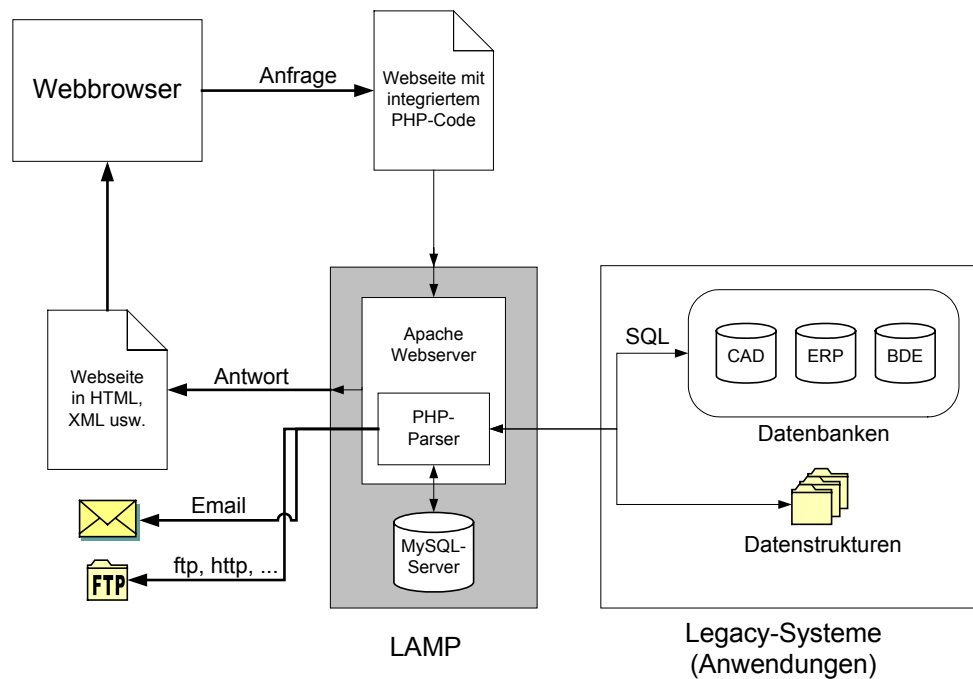


Abbildung 5.21: LAMP-Integrationsschema

Auch Webservices, die auf der Nutzung standardisierter Softwarekomponenten und Verfahren beruhen und mit deren Hilfe sich Daten und Informationen austauschen lassen, werden von PHP unterstützt. Etabliert sind neben den Kommunikationsprotokollen TCP/IP auch das bereits genannte Datenaustauschformat XML, das Protokoll SOAP, das unter Verwendung von XML zur Kommunikation von verteilten Anwendungen über das Internet dient, der Standard zur Beschreibung von Schnittstellen **WSDL**¹³⁴ und schließlich das Webservice-Verzeichnis **UDDI**¹³⁵. Technologisch wird damit durch PHP in Verbindung mit Apache die Kommunikationskomponente gebildet, über die die Kommunikation zwischen den Anwendungen stattfindet und über die Nachrichten für Webservices transportiert werden.

Durch den gemeinsamen Einsatz der LAMP-Komponenten entsteht somit eine Integrationslösung, die die wesentlichen Kerneigenschaften eines webbasierten EAI-Systems besitzt. Zwar ist in Anlehnung an die Eigenschaften der dreischichtigen EAI-Referenzstruktur aus Abbildung 3.6 eine eindeutige Zuordnung der LAMP-Komponenten auf die drei Schichten nicht möglich, allerdings auch nicht erforderlich. Vereinfacht lässt sich sagen, dass die Prozessschicht gemeinsam durch PHP und MySQL abgebildet wird, die

¹³⁴ Web Services Description Language

¹³⁵ Universal Description, Discovery and Integration

Kommunikationsschicht durch Linux, PHP und Apache sowie die Protokoll-Adapterschicht durch Linux und PHP.

Über die rein inhaltliche Abbildung der EAI-Referenzstruktur hinaus besitzt die LAMP-Umgebung weitere Merkmale, die für Integrationssysteme bedeutsam sind. Wie bei einigen in Kapitel 4 beschriebenen EAI-Lösungen auch stehen beispielsweise mit dem Verzeichnisdienst LDAP systemübergreifende Authentifizierungs- und Verschlüsselungsmechanismen zur Verfügung, die eine reibungslose Integration in eine heterogene Infrastruktur ermöglichen. LDAP bietet die Möglichkeit, unterschiedliche Anwendungssysteme über eine gemeinsame Authentifizierungsinstanz zusammenzuführen und auf Authentifizierungsebene zu integrieren. Dadurch können Anwender für alle benötigten Anwendungssysteme dieselben Anmelde- bzw. Accountinformationen verwenden, wodurch einerseits Sicherheit und Administration sowie andererseits die Zufriedenheit auf Anwenderseite deutlich verbessert werden. Ein pragmatischer Grund, der für die Verwendung von LAMP als EAI-System spricht, ist dagegen vergleichsweise trivial: in mittelständischen Unternehmen ist vielfach bereits eine LAMP-Umgebung vorhanden, so dass die infrastrukturellen Systemkosten keinen oder nur einen als sehr gering zu bewertenden Kostenfaktor darstellen.

Für die Eigenschaften des Load Balancing, Failover, Monitoring, Recovery-Fähigkeit, Tracing und Debugging existieren unter Linux betriebssystemseitige Lösungsansätze, die gemeinsam mit den Funktionalitäten der Protokollierung, Performance, Skalierbarkeit und Zuverlässigkeit auf die Alltagstauglichkeit eines Systems abzielen. Sie sind daher nicht EAI-eigen, sondern als elementar für geschäftskritische Softwaresysteme anzusehen und werden durch die Teilsysteme Linux, Apache, MySQL und PHP erfüllt.

Zusammengefasst erfüllt die LAMP-Technologie in weiten Teilen die umfangreichen Anforderungen an eine Integrationsplattform, wie sie in Kapitel 3 beschrieben wurden, und durch die Unterstützung für Webservices ist sie in besonderer Weise für den Einsatz als webbasiertes EAI-System geeignet. Strukturell ergibt sich dabei eine Hub & Spoke-Architektur mit einem zentralen EAI-Server als (Daten-)Integrationskern (vgl. Abbildung 5.22), und die Benutzerschnittstelle ist ein Webbrowser (Anwendungs-Frontend), in dem die Daten ausgegeben werden.

Damit kann nunmehr das Konzept für die Anwendungsintegration auf der Basis von LAMP mit Blickrichtung auf das SCM exakter beschrieben werden.

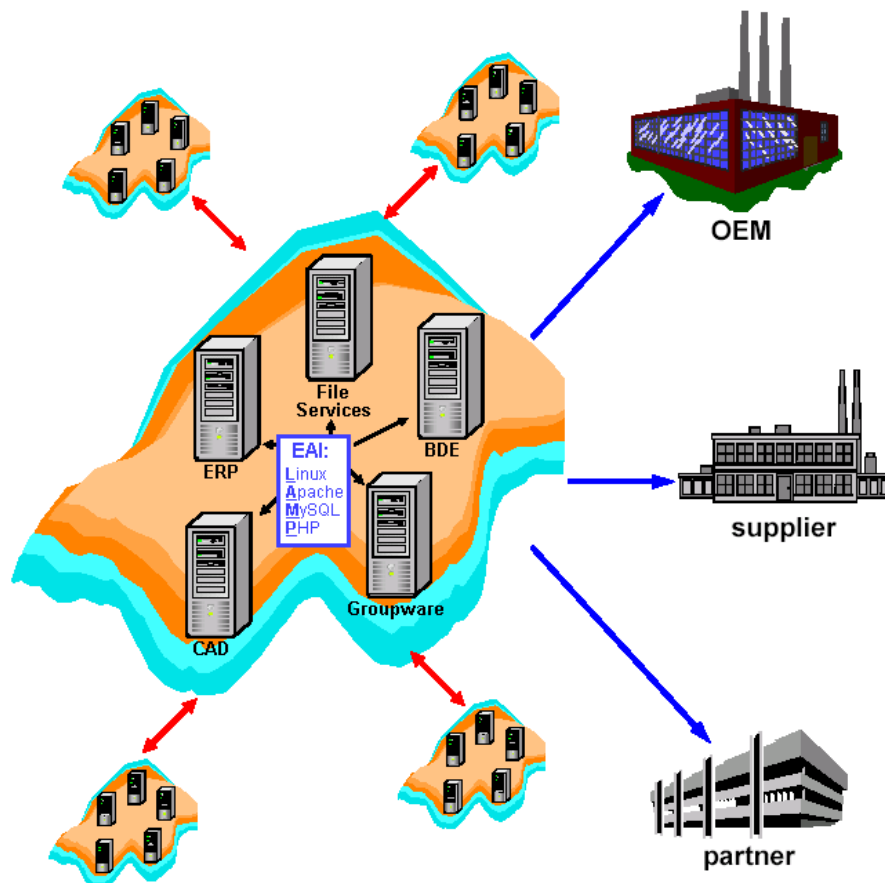


Abbildung 5.22: EAI-Systemstruktur mit LAMP

Innerhalb des Supply Chain Managements wird grundsätzlich unterschieden zwischen interner und externer Supply Chain. Während bei internen Supply Chain-Prozessen der Fokus auf die unternehmensinternen Vorgänge und Systeme gerichtet ist, treten bei den externen SCM-Prozessen Schnittstellen zu Kunden, Lieferanten und sonstigen Partnern auf. Bei der internen Supply Chain ist daher eine Verbindung zwischen den internen Unternehmensanwendungen erforderlich, im Falle der externen Supply Chain ist es eine Verbindung zwischen den Anwendungs- und Prozesssystemen unterschiedlicher Unternehmen, die über Unternehmensgrenzen hinweg verläuft und die daher als B2B-Anwendung bezeichnet wird. Sie benötigt das Internet als Träger- bzw. Kommunikationssystem, wodurch unter Anwendung von SCM-spezifischen Konzepten die Geschäftsprozesse der Unternehmen gekoppelt bzw. integriert werden und der Prozessablauf automatisiert wird.

In der Literatur wird für die Beschreibung der Supply Chain Management-Prozesse häufig von dem **SCOR**¹³⁶-Modell ausgegangen [3, 83, 114]. Innerhalb des SCOR-Modells werden

¹³⁶ Supply Chain Operations Reference-Modell

drei Prozesstypen unterschieden: Planungs-, Ausführungs- und Infrastrukturprozesse. Während die Infrastrukturprozesse Informationen und Beziehungen über den Ausführungs- und Planungsprozess vorbereiten und verwalten, wird in den Ausführungsprozessen der Zustand des Materials oder der Waren verändert. In den Planungsprozessen werden schließlich die Aktivitäten festgelegt, um Angebot und Nachfrage in Einklang zu bringen. Hier sind die Vertriebs-, Beschaffungs- und Produktionsplanung angesiedelt, die in der Abbildung 5.23 zur Kategorisierung der Teilbereiche des Supply Chain Managements dargestellt sind.

Innerhalb der Teilbereiche des SCM existieren strategische Fragestellungen wie

- Sourcing-, Standort- und Distributionsstrategien
- **PDM**¹³⁷ und
- **SCP**¹³⁸, aber auch technische bzw. operative wie
- **VMI**¹³⁹ aus dem Bereich des
- Electronic Procurement bzw.
- Electronic Commerce,
- PPS,
- Tracking and Tracing oder
- Abrufverfahren via JIT.

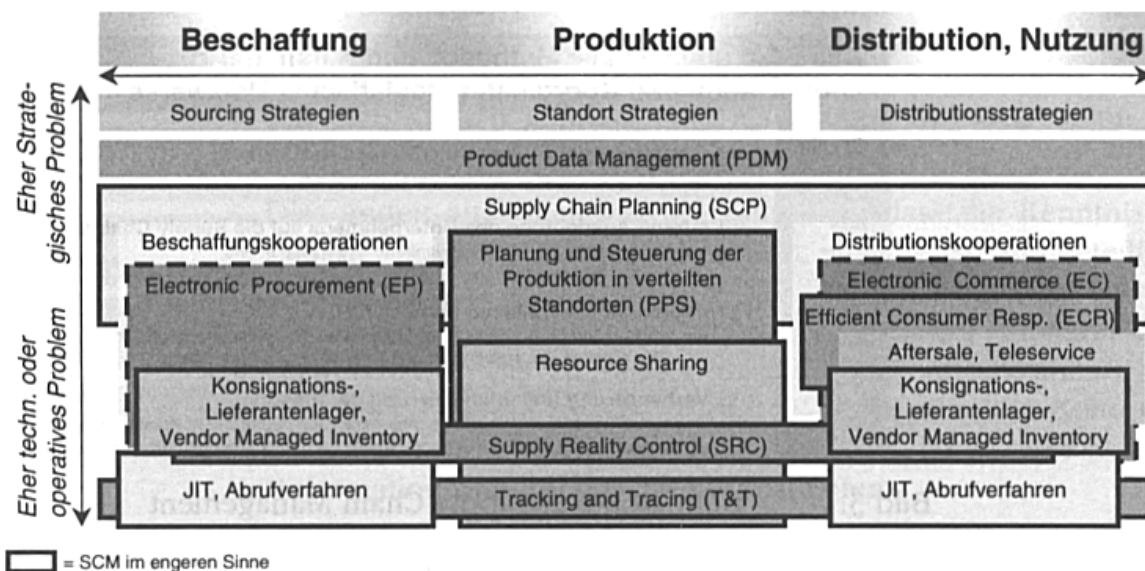


Abbildung 5.23: Teilbereiche des Supply Chain Managements [3]

137 Product Data Management

138 Product Data Management
Supply Chain Planning

139 Supply Chain Planning
Vendor Managed Inventory

Unter diesen Gesichtspunkten wird nun die Situation in einem zur Automobilindustrie gehörenden mittelständischen Unternehmen betrachtet, wie sie bereits in Kapitel 2.8, Abbildung 2.12 dargestellt ist. Das Unternehmen ist dadurch geprägt, dass in einer heterogenen Software- und Systemstruktur zur Abbildung der internen Prozesse unterschiedliche IT- und Softwaresysteme eingesetzt werden. Es existieren organisatorisch voneinander getrennte Abteilungen (Personal-, Rechnungs- und Qualitätswesen, Versand, Entwicklung und Vertrieb, Fertigungsplanung sowie Einkauf und **EDV**¹⁴⁰) mit jeweils abteilungsspezifischen Softwareanwendungen, die integriert werden müssen.

Bei genauer Betrachtung der Beziehungen des Unternehmens zu Kunden und Lieferanten existieren teilweise isolierte Geschäftsprozesse, innerhalb deren der Austausch von Prozessdaten mit punktuellen Verbindungen anwendungsspezifisch ermöglicht wird. Eine Verbindung zu nicht direkt angebundenen Systemen existiert dabei nicht, weder intern noch extern.

Durch die Anwendungsintegration mit LAMP als zentralem Integrationselement wird nun die Verbindung zwischen den einzelnen Abteilungen und deren Anwendungen bzw. der zugrunde liegenden Unternehmensprozesse hergestellt, woraus schließlich das in der Abbildung 5.24 dargestellte EAI-LAMP-Szenario resultiert. Zentrales Integrationselement ist dabei der EAI-LAMP-Server, der die benötigte Integrationsfunktionalität zur Verfügung stellt. Diese Integrationsfunktionalität liegt darin, dass die Daten aus den abteilungsspezifischen Softwaresystemen mittels PHP zunächst extrahiert werden, ggf. konvertiert und entweder mit Unterstützung des Apache Webserver über ein Webfrontend ausgegeben, als Nachricht versendet oder direkt in eine MySQL-Datenbank geschrieben werden.

Die Berechtigungsstrukturen für die an den externen Stellen erforderlichen Daten- und Datenbankzugriffe werden dabei über eine MySQL-Datenbank zugewiesen und beim Programmaufruf gegen diese Datenbank geprüft. Technologisch werden hierbei verschiedene Kommunikationsstrukturen eingesetzt, die über Kommunikationselemente wie

- http,
- ftp bzw.
- SQL oder
- XML

¹⁴⁰ Elektronische Datenverarbeitung

abgebildet werden. Auf diese Weise werden die internen Unternehmensbereiche integriert und es entsteht durch die LAMP-Struktur ein zentrales Kommunikationselement, über das auch die Kommunikation zu Kunden und Lieferanten erfolgt. Entlang dieser Kommunikationskanäle werden die für das interne bzw. externe Supply Chain Management benötigten Prozesse abgebildet.

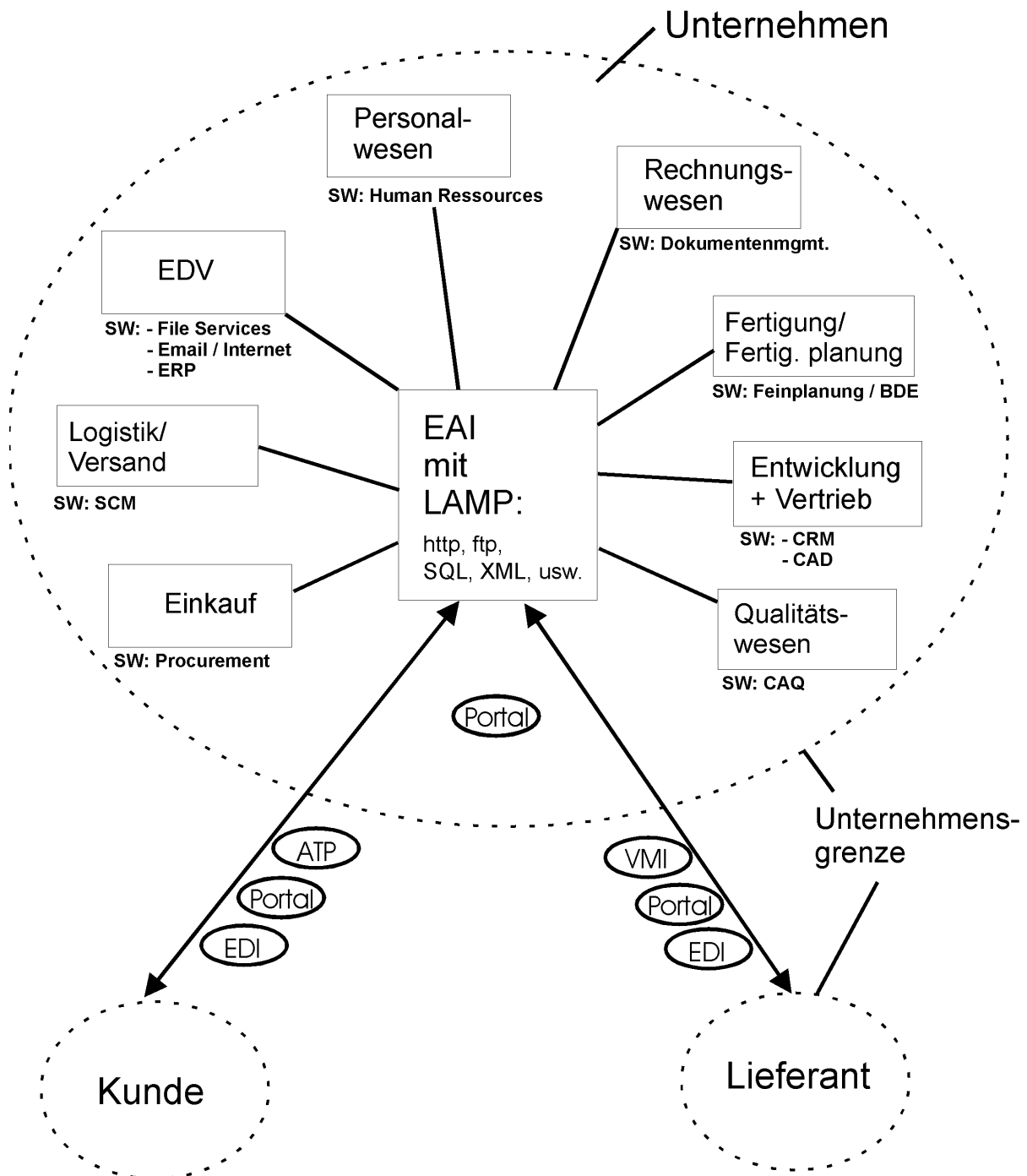


Abbildung 5.24: Situation der Beziehungen im EAI-LAMP-Szenario

Exemplarisch sind im EAI-LAMP-Szenario vier SCM-Prozesse dargestellt: VMI, ATP, Portal und der Prozessdatenaustausch via EDI. Unter Berücksichtigung der Integrationsanforderungen dieser Prozesse werden ausgehend von der Eignung der LAMP-Umgebung als EAI-Tool diese Teilbereiche des Supply Chain Managements im Folgenden näher betrachtet, da diese spezifisch für das mittelständische Automotive-Umfeld sind. Dabei werden die Technologien konzeptionell beschrieben, die für eine Integration mit LAMP erforderlich sind.

5.5 Detailkonzepte

Bei den nachstehend aufgeführten und in der Abbildung 5.24 dargestellten SCM-Prozessen in der Automobilindustrie handelt es sich um die folgenden Szenarien, die der internen bzw. externen Supply Chain zuzuordnen sind:

1. Umsetzung eines VMI, auch Konsignationslager genannt, (externe Supply Chain),
2. Anfragen der Art ATP – Available To Promise (interne / externe Supply Chain),
3. Umsetzung einer Portallösung (interne / externe Supply Chain) und
4. Prozessdatenaustausch / EDI (interne / externe Supply Chain).

Die für die Konzeption jeweils zugrundegelegte technologische Basis beruht auf der Verwendung der Integrationsfunktionalität, die durch die LAMP-Umgebung zur Verfügung gestellt wird und die im vorherigen Kapitel beschrieben wurde.

5.5.1 VMI

Innerhalb des SCM stellt das Vendor Managed Inventory einen B2B-Prozess dar. Vereinfacht formuliert beschreibt das VMI das Verwalten und Auffüllen von Lagerbeständen durch Lieferanten. Damit wird der Lieferant verantwortlich für den im Lager vorhandenen Lagerbestand. Das VMI wird von vielen Experten seit geraumer Zeit als eine der größten Herausforderungen angesehen, mit der sich die Industrie momentan auseinander zu setzen hat. Zudem gewinnt VMI immer mehr an Bedeutung, wenn es darum geht, kosteneffizient und dennoch kundenorientiert einzulagern. Über automatisierte Internet-Prozesse, beispiels-

weise Email oder Internet-Login, erhalten die Lieferanten Zugang zu den dispositiven Daten. Dadurch überwacht der Lieferant das Lager seines Kunden. Im Falle einer Bestandsunterdeckung erfolgt seitens des Lieferanten ohne eigenes Zutun eine Anlieferung der benötigten Teile, wodurch optimale Auffüllmenge und Auffüllzeitpunkt erreicht werden. Periodisch werden die Verbräuche in Rechnung gestellt. Die wesentlichen Vorteile eines VMI gegenüber einem selbstverwalteten Lager sind

- Verringerung der Anzahl manueller Bearbeitungsstellen (Beleganfall)
- Reduzierung der Prozesskosten
- Verbesserung der Supply Chain
- Verkürzung der Durchlaufzeit / schnellere Auftragsabwicklung
- Verbesserte Materialverfügbarkeit und Lieferfähigkeit
- Verringerung der Lagerkapazität
- Bessere Kapazitätsauslastung / Glätten von Bedarfsschwankungen
- Optimierte Versandkosten und -zeiten durch z. B. verbesserte Tourenplanung.

Von besonderer Bedeutung im Rahmen des Asset Management und Working Capital Controlling sind die Lagerbestände, die mit 34 Prozent den größten Anteil am Umlaufvermögen darstellen [115]. Das hierdurch gebundene Kapital, durchschnittlich 13 bis 14 Prozent des Umsatzvolumens deutscher Unternehmen, ist ein wesentlicher Ansatzpunkt einer nachhaltigen Optimierung der wirtschaftlichen Situation der Unternehmen. Da Bestände als Teil des Umlaufvermögens meist über Fremdkapital finanziert werden, mindern Vorräte durch Zinsaufwendungen und Lagerungskosten den Gewinn. Gleichzeitig binden Bestände auch Kapital, die Rentabilität eines Unternehmens wird also durch Vorratshaltung in zweifacher Weise beeinflusst. Bestände mindern unmittelbar durch Zinsaufwendungen und andere Bestandskosten den betrieblichen Erfolg und erhöhen den Ressourceneinsatz, in diesen Fällen durch Erhöhung des gesamten Vermögens oder durch Erhöhung des eingesetzten Kapitals. Bestände sind ein geeigneter Hebel für eine nachhaltige Ergebnisverbesserung der Unternehmen. Auch nach Andersen Consulting ist der Payback von vielen VMI-Investitionen beträchtlich. Schätzungen ergeben, dass mit Hilfe von VMI das Umlaufvermögen um 20 bis 30% erniedrigt werden kann. Zudem sind die Investitionen in weniger als einem Jahr amortisiert, die Effizienz des Bestandsmanagements aber über viele Jahre drastisch verbessert.

Exemplarisch für die Wirtschaftlichkeit von VMIs ist die Verkürzung der Auftragsabwicklung anhand der Abbildung 5.25 beschrieben. Durch die zeitnahe Kenntnis von Lagerbestand bzw. Lagerbestandsunterdeckungen, automatisiert erzeugten EDI-Bestell- und Bestätigungs-

vorgängen sowie Versand- und Zahlungsprozessen kann an den beispielhaften Distributionsprozessen von Gebrauchsgütern neben niedrigeren Lagerbeständen und weniger administrativem Aufwand eine Verkürzung der Auftragsabwicklung um bis zu 12 Tage eintreten [116], was quasi einer Halbierung der Zeit entspricht, die für die Auftragsabwicklung insgesamt benötigt wird.

Hard Goods Distribution Processing

Order Process with VMI		Order Process without VMI	
Process	Days	Process	Days
• Review current inventory levels pulling files from distributor to vendor	1 to 2	• Review current inventory	1 to 2
• Vendor sends EDI document verifying data	1	• Check system data for accuracy	1
• Vendor puts together anticipating order for distributor	1 to 2	• Review all branch inventories	1 to 2
• Vendor sends advanced shipping notice to vendor via EDI	1	• Put together purchase order for vendor	1 to 2
• Distributor verifies order; sends back-order acknowledgment via EDI	1	• Mail order to vendor	4
• Vendor picks order	2 to 3	• Vendor receives order and verifies it with distributor	2 to 3
• Vendor ships order	4	• Vendor picks order after talking with distributor	2 to 3
• Distributor receives product without check-in and verification	1	• Vendor ships order	4
• distributor acknowledges via EDI receipt of product	1	• Distributor receives product and checks order	1 to 2
• Vendor paid via electronic funds from distributor's bank	1	• Distributor calls vendor to discuss discrepancies	1 to 2
		• Distributor writes check and mails to vendor	4
Total	14 to 17	Total	22 to 29

(nach Emigh)

Abbildung 5.25: Verkürzung der Auftragsabwicklung mit und ohne VMI [116]

Den Vorteilen eines VMI stehen jedoch hohe Implementierungskosten gegenüber, die nach Andersen Consulting bei rund 150.000 US \$ zur Überwachung eines kleineren Lagers beginnen, vorausgesetzt die Ermittlung des Bestandsniveaus und die Berechnung der Bedarfsprognosen erfolgen nur in größeren Zeitabständen. Hingegen müssen für ein ausgereiftes System, das mittels Sensoren täglich aktuelle Prognosen errechnet, einmalige Kosten in Höhe von bis zu 500.000 US \$ eingeplant werden.

Auf Prozessebene stellt sich die Abbildung eines VMI vereinfacht wie in der Abbildung 5.26 skizziert dar. Während die Lagerbestandsführung eines Kunden vor der Implementierung eines VMI von diesem selbst durchgeführt werden muss, indem Teilprozesse wie Lagerbestandsprüfungen und Bestellvorgänge durchgeführt werden, erfolgen diese Schritte

nach der Umsetzung des VMI für den Kunden automatisiert und auf der Seite des Lieferanten.

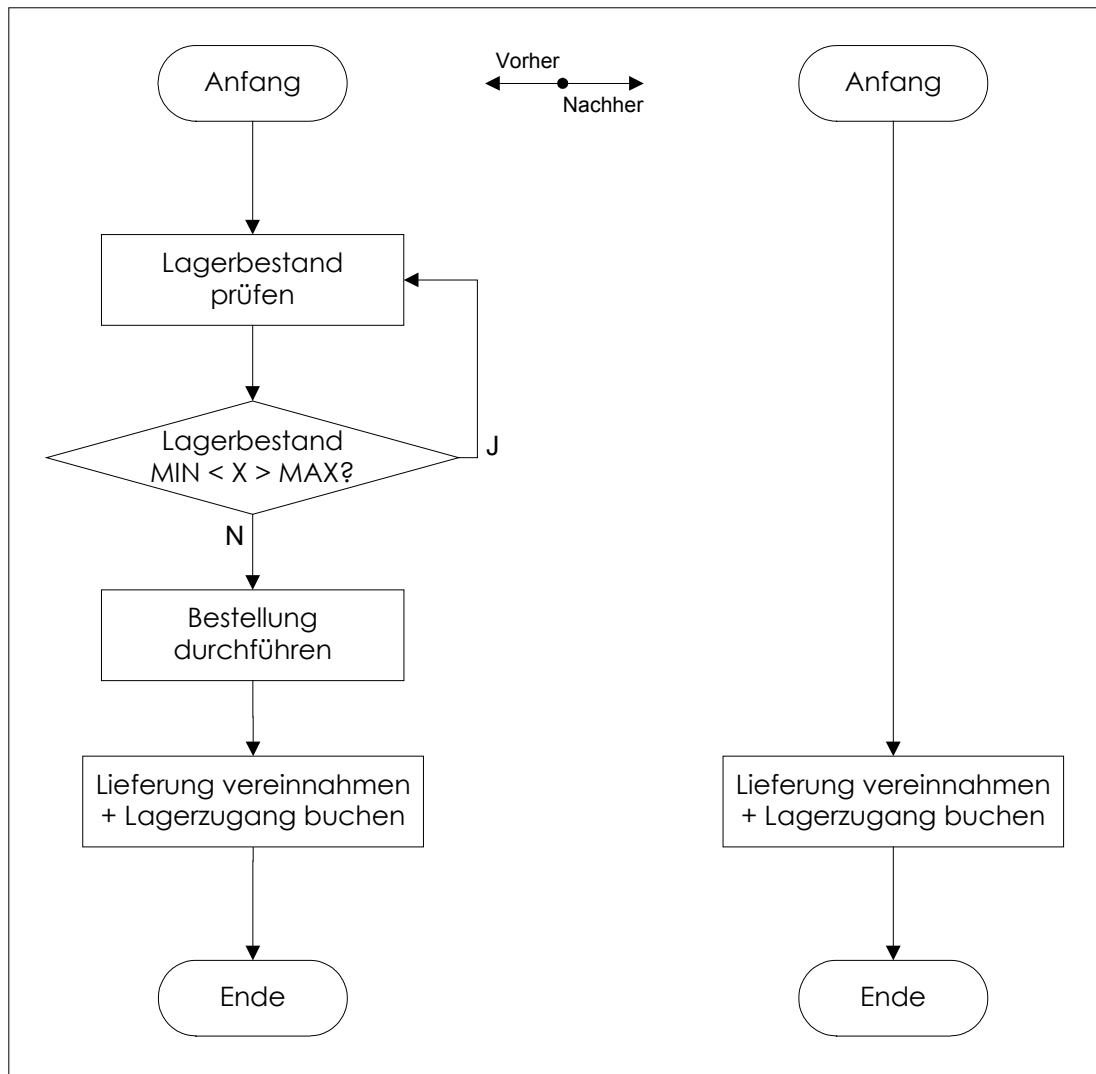


Abbildung 5.26: Prozessstruktur beim VMI

Damit vereinfacht sich die Situation auf der Kundenseite, und der Lieferant erhält darüberhinaus einen permanenten Einblick in die aktuellen und von ihm zu verantwortenden Lagerbestände. Er wird dadurch in die Lage versetzt wird, zeitnaher und mit größeren dispositiven Freiheitsgraden die betreffende Ware an seinen Kunden auszuliefern.

Einen günstigen, schnell implementierbaren und skalierbaren Ansatz für die Umsetzung eines VMI liefert die EAI-LAMP-Umgebung. Dafür werden zunächst die am VMI beteiligten Systeme in einer mittelständischen Systemumgebung betrachtet.

Innerhalb des Supply Chain Managements ist das VMI bei den **APS**¹⁴¹ anzusiedeln. APS-Systeme beziehen ihre Daten häufig aus vorhandenen ERP-Systemen. Durch wechselseitige Systemverbindungen und entsprechende Schnittstellen können zwar relevante Daten aus den ERP-Systemen in die APS-Systeme eingespielt, dort verarbeitet und die so ermittelten Ergebnisse anschließend wieder an das ERP-System zurückgegeben werden [117], eine Verbindung zu externen (Lieferanten-)Systemen (hier: Webbrowser) ist damit jedoch noch nicht hergestellt. Deshalb müssen die Daten, die für ein VMI erforderlich sind, aus dem ERP-System extrahiert und anschließend in einer Internet-Umgebung zur Verfügung gestellt werden, d. h. sie müssen via Internet abrufbar bzw. darstellbar sein. Hier liegt insofern ein Integrationsproblem innerhalb der externen Supply Chain zwischen ERP- und Internet-Plattform vor.

Die Integrationsanforderungen werden durch die operativen Daten auf der einen Seite und den beteiligten Anwendungen der Anwender-Sicht auf der anderen Seite definiert. Bei genauerer Betrachtung finden sich auf der Datenseite Datenbanken oder darauf basierende Anwendungen wieder, bei Legacy-Systemen können mitunter auch Flat-Files existieren, in denen die relevanten Daten strukturiert in Form von Dateien vorhanden sind. Ausgehend von den Daten aus dem ERP-System ist über Datenbankserver und ggf. Metadatenstrukturen die Anwendungsintegration über eine Web-Anwendung bzw. einen Webserver durchzuführen, an deren Ende die Anwender-Sicht auf die operativen Daten steht.

Dieses Problem ist unter Verwendung von LAMP dadurch lösbar, dass mit PHP die erforderliche Datenintegration durchgeführt wird und mit Apache die Ausgabe der über PHP- bzw. HTML-Code aufgebauten Webseiten erfolgt. Schematisch ergibt sich damit in Anlehnung an das allgemeine LAMP-Integrationsschema aus Abbildung 5.21 das konkrete Integrationsszenario, das in der Abbildung 5.27 dargestellt ist. Im ersten Schritt erfolgt die Anfrage des Webbrowsers, der die Lagerbestandsdaten, die im ERP- oder Lagerhaltungssystem abgelegt sind, in Form eines PHP-Seitenaufrufs anfordert. Der Apache-Webserver übergibt diesen Aufruf an den PHP-Parser, der einen Datenbankzugriff auf die benötigten Lagerbestandsdaten des ERP-Systems durchführt. Zur Selektion der erforderlichen Datensätze wird innerhalb des PHP-Skriptes die Datenbanksprache SQL verwendet. Auf der LAMP-Seite wird dazu je nach Anforderung ein nativer Datenbank-Connect oder ein ODBC-Zugriff auf die externen ERP-Daten (hier: Lagerbestandsinformationen) verwendet, wozu die PHP-Funktionen `mysql_connect` für beispielsweise nativ unterstützte MySQL-Datenbanken oder `odbc_connect` für via ODBC angesprochene Datenquellen dienen. Die Lagerbestandsinformationen werden dadurch aus

¹⁴¹ Advanced Planning System

dem ERP-System gelesen und entweder in einem Zwischenschritt in einer MySQL-Datenbank gespeichert oder direkt über den Webserver Apache in Form von Webseiten beispielsweise als HTML- oder XML-Datenstrom an das Anwendersystem (Webbrowser) ausgegeben.

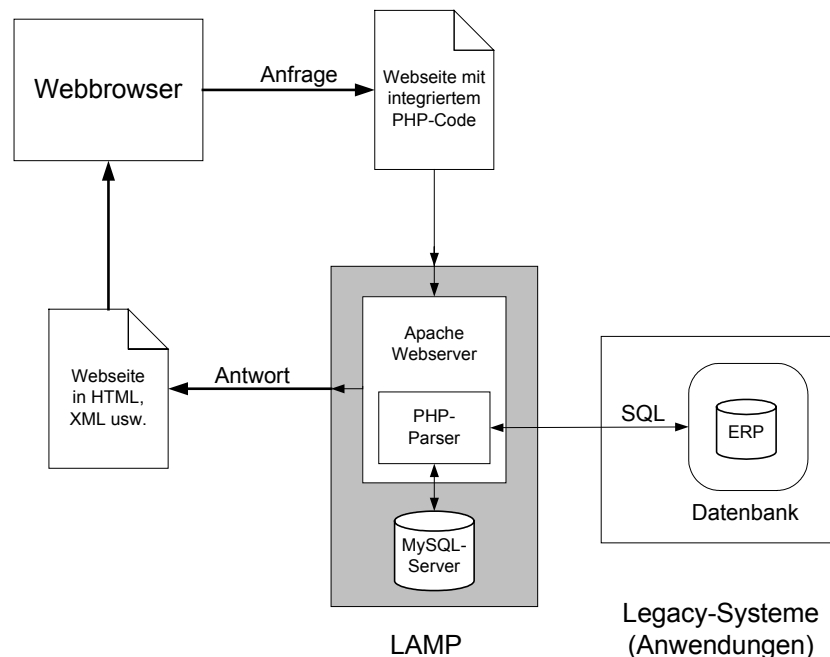


Abbildung 5.27: LAMP-Integrationselemente beim Vendor Managed Inventory

Dadurch erfolgt die für den Anwender transparente Weitergabe der relevanten Informationen von einem für ihn fremden System zum von ihm verwendeten Internet-Browser. Alternativ dazu ist auch der automatisierte Versand der Informationen möglich, beispielsweise per Email über die Mail-Funktion oder als XML-Datenstrom in ein Zielsystem des Lieferanten über die XML-Unterstützung von PHP.

Durch die Verwendung von MySQL können bei Bedarf die bereitgestellten Daten nicht nur zwischengespeichert, sondern auch aus den einzelnen Datenquellen konsolidiert in Form eines Data Warehouses abgelegt werden. Dies kann aus Performancegründen erforderlich werden, beispielsweise wenn große Datenmengen verarbeitet werden müssen oder mehrere ERP-Systeme mit spezifischen Teilfunktionalitäten beteiligt sind, aus denen Informationen extrahiert und zusammengeführt werden müssen. In diesem Fall erfolgt die Ausgabe der einzelnen PHP-Daten(bank)-Connects in eine MySQL-Datenbank, um die Ausgabe der benötigten Daten zu einem späteren Zeitpunkt mittels HTML oder XML in performanter Form zur Verfügung stellen zu können.

Damit ist die technologische Verknüpfung von ERP- und Anwendersystem in Form einer webbasierten Anwendungsintegration mit LAMP beschrieben, die die Basis für den SCM-Prozess und das in Echtzeit abgebildete VMI darstellt. In Abhängigkeit der funktionalen Anforderungen an das VMI (z. B. automatisierter Email-Versand bei Unterschreiten der Minimal-Bestände oder automatisierter Rechnungsversand) sind nun beispielsweise über die Benachrichtigungsfunktionen von PHP die weiteren zusätzlich erforderlichen Eigenschaften des VMI abbildbar.

5.5.2 ATP

Über die Verfügbarkeit von Informationen über Lagerbestände innerhalb eines VMI hinaus ist im Rahmen der Beziehungen zwischen Kunden und Lieferanten heute die Angabe definierter Liefertermine ("Order Promising") von größter Bedeutung. Kunden wollen immer schneller und zuverlässiger wissen, ob zu einem vorgegebenem Termin geliefert werden kann. Auf der anderen Seite benötigt auch der Lieferant sofort Informationen darüber, was ein eingehender Auftrag hinsichtlich Lieferbarkeit und Fertigungskapazität für ihn bedeutet. Hieraus ist der Begriff des ATP entstanden. Dabei wird unter Berücksichtigung von verfügbarem Enderzeugnisbestand und bereits eingeplanten Fertigungsaufträgen bei einer Kundenanfrage bestimmt, ob der gewünschte Liefertermin erfüllt werden kann oder welcher frühest mögliche Liefertermin sich ergibt. Eine Unterfunktion von ATP stellt **CTP**¹⁴² dar. Hierbei werden die für diesen Auftrag benötigten Fertigungsaufträge unter Berücksichtigung der bereits verplanten Kapazität und der Materialien (zusätzlich) eingeplant. Das Ziel ist, automatisierte Bestands- und Fertigungsinformationen in Echtzeit an die Kunden weiterzugeben [92]. ATP-Szenarien treten innerhalb von B2B- oder **B2C**¹⁴³-Prozessen auf.

Das Grundprinzip des ATP-Prozesses ist exemplarisch in der Abbildung 5.28 dargestellt. Ohne die Anwendungsintegration mit LAMP ist der erste manuell durchzuführende Teilprozess die Anfrage eines Kunden, ob die von ihm gewünschte Ware lieferbar ist. Dies geschieht zumeist auf telefonischem Wege mit den hieraus resultierenden Nachteilen (Erreichbarkeit, Verbindlichkeit). Daraufhin wird auf der Lieferantenseite eine Verfügbarkeits- bzw. Lieferbarkeitsprüfung durchgeführt. Ergibt die Prüfung, dass die Ware lieferbar ist, so erhält der Kunde telefonisch eine entsprechende Lieferbarkeitszusage, ansonsten bekommt er eine Information darüber, dass die Lieferbarkeit nicht gegeben ist.

¹⁴² Capable To Promise

¹⁴³ Business To Customer

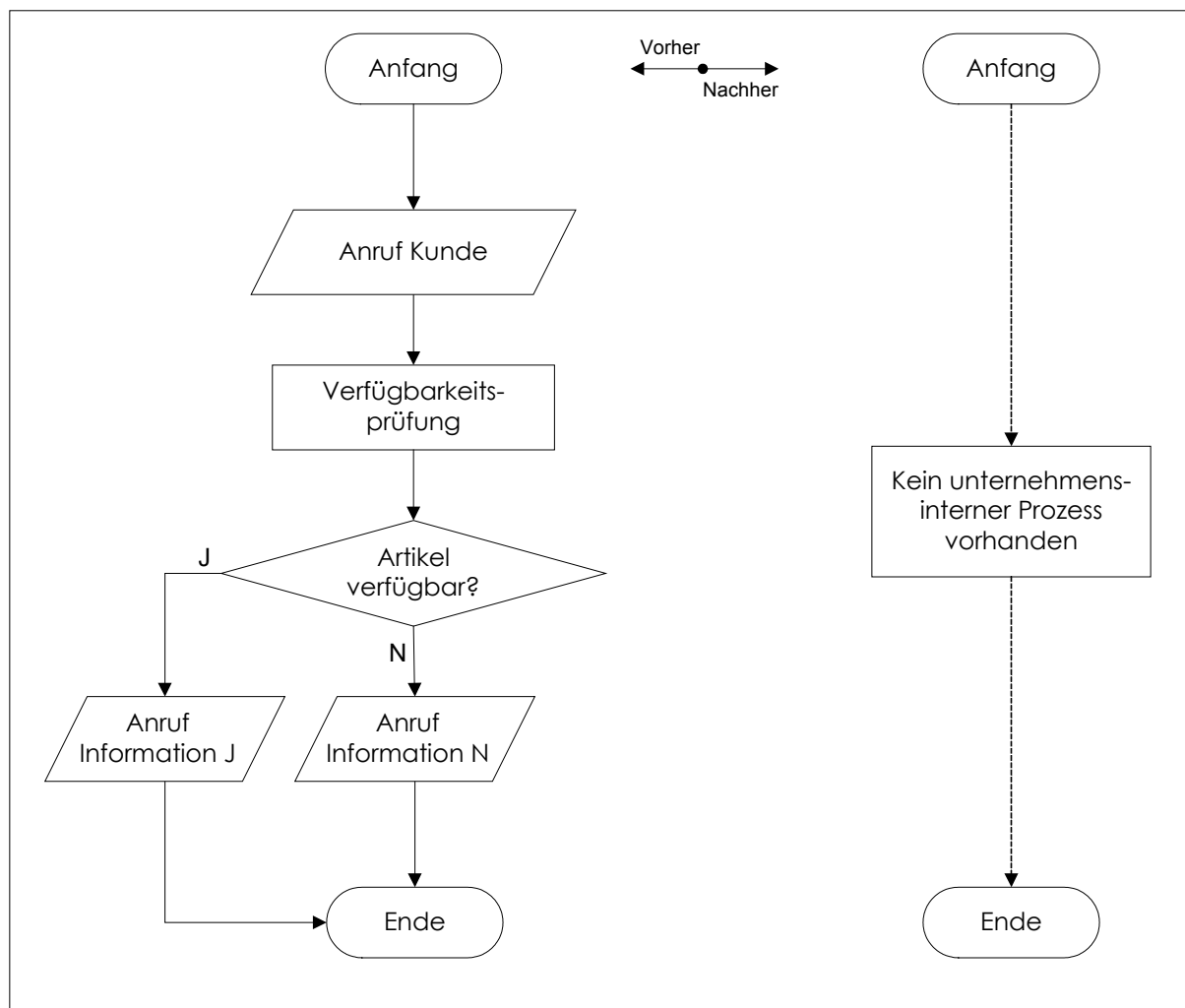


Abbildung 5.28: Prozessstruktur bei ATP

Durch die Abbildung des Prozesses mit Hilfe der LAMP-Umgebung ist der Kunde in der Lage, die von ihm benötigten Informationen ohne manuelle Einbindung des Lieferanten direkt aus dem System des Lieferanten abzufragen. Dadurch werden auf der Lieferantenseite die zuvor beschriebenen Teilprozesse komplett über das LAMP-System abgebildet, so dass aus unternehmensinterner Sicht kein manueller Teilprozess mehr erforderlich ist. Gleichzeitig sind die Informationen in verlässlicher Form und zu jeder Zeit von dem Kunden abrufbar.

Beim ATP-Szenario sind hinsichtlich der Integrationsanforderungen mehrere Integrationsfälle denkbar. Entweder stammen Bestands- und Fertigungsdaten aus einem gemeinsamen System (z. B. ERP-System) oder aber aus unterschiedlichen, voneinander getrennten Systemen, beispielsweise ERP-System und Feinplanungs- bzw. BDE-System. Während im ersten Fall der technologische Integrationsumfang ähnlich wie beim VMI aus zwei Systemen besteht, kommt im zweiten Fall ein weiteres System hinzu. Es ist dann erforderlich, über den

LAMP-Integrationsserver diese drei Systemumgebungen anzubinden, was in der Abbildung 5.29 dargestellt ist

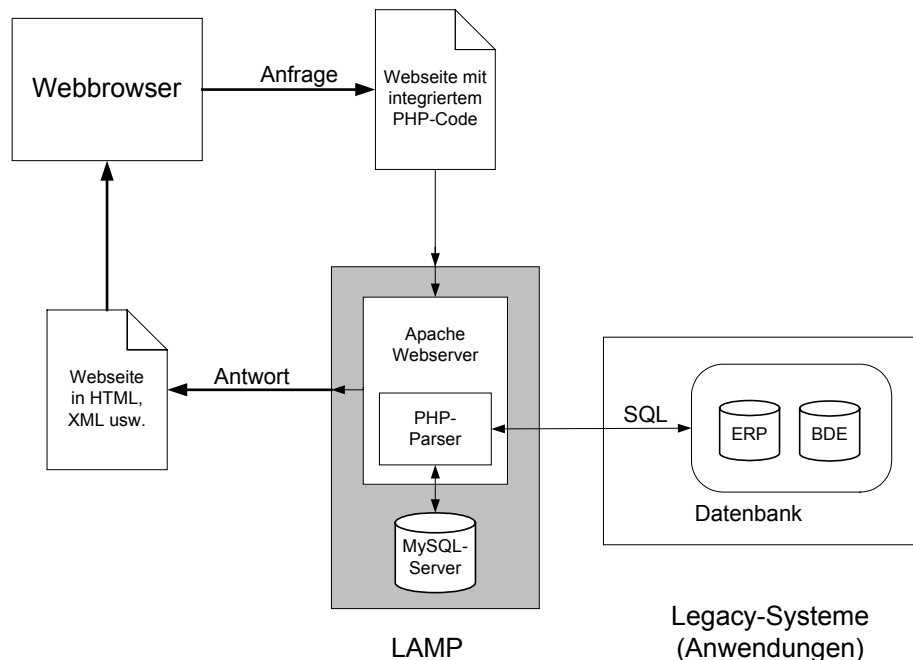


Abbildung 5.29: LAMP-Integrationselemente bei ATP

Grundsätzlich geschieht die Anbindung über den gleichen Ablauf wie beim VMI. Es werden über das ERP-System, das die Lagerbestandsinformationen verwaltet, zum einen die aktuellen Lagerbestände abgefragt und zum anderen über das Feinplanungssystem die geplanten Fertigungsaufträge mit Start- und Endzeit unter Berücksichtigung des jeweiligen Planungshorizonts. Dies geschieht via PHP entweder über einen nativen Datenbankzugriff auf die Datenbanken der beiden Systeme oder über einen ODBC-Zugriff, also wiederum unter Verwendung der Funktionen `odbc_connect` oder entsprechender PHP-Datenbankfunktionen. Mittels SQL werden die Daten bzgl. des Lagerbestandes und der Fertigungsaufträge innerhalb eines PHP-Skriptes zusammengeführt und anschließend strukturiert über den Webserver oder in eine MySQL-Datenbank ausgegeben. So entsteht eine Übersicht über die vorhandenen Lagerbestände einerseits und die geplanten oder zum Abfragezeitpunkt aktuell vorhandenen Fertigungsaufträge andererseits, woraus auf Anwenderseite die Lieferbarkeit eines Artikels ersichtlich wird. An dieser Stelle können nun noch weitere Informationen dargestellt werden, beispielsweise über aktuelle Lieferaufträge, die mengenmäßig zu einer Verringerung der tatsächlich verfügbaren Bestände führen, oder sich in der Auslieferung befindende Liefermengen. Diese Anforderungen sind im Funktionsumfang des ATP-Prozesses zu definieren und im Anschluss daran wie oben beschrieben umzusetzen.

5.5.3 Portallösung

Eine umfassende Anwendung für viele Arten von Unternehmensinformationen stellen Internetportale dar. Diese unterscheiden sich von einer einfachen Homepage dadurch, dass personalisierte Informationen für den Benutzer bereitgestellt werden. Eine Portalanwendung ist also eine B2C-Anwendung auf Endkundenebene bzw. eine B2B-Anwendung auf Zulieferebene im Sinne von E-Procurement und erlaubt z. B. einem Kunden, direkt aus den Datensystemen des Lieferanten Informationen über sich selbst, seine Aufträge, seine Rechnungen usw. zu erhalten. Hinter der Portal-Oberfläche läuft ein Integrationsprozess ab, um die produktiven Informationen aus den Lieferantensystemen herauszulesen und dem Web-Frontend zu übergeben. Eine Portal-Anwendung ist daher eine Internet-Anwendung und die Anbindung der Lieferantensysteme ist eine EAI-Anwendung. Portallösungen bilden auch die Grundlage für beispielsweise Publishing im Internet, Produkt-Präsentation oder einen personalisierten Produktkatalog. Sie dienen der Bereitstellung von Informationen und der Prozessunterstützung sowohl im unternehmensinternen als auch im -externen Bereich und beziehen ihre Informationen aus einer Vielzahl von unterschiedlichen Systemen. Damit stellen Portale eine Informationsplattform dar, die zahlreiche manuelle Informationsprozesse mit individuellen Informationsbedarfen, in denen einzelne Informationen aus Einzelsystemen zusammengetragen und aufbereitet werden müssen, durch zentral installierte Automatismen ersetzen. Dies ist schematisch in der Abbildung 5.30 dargestellt, in der durch die LAMP-Umgebung die benötigte Integrationsfunktionalität zur Verfügung gestellt wird.

An Portallösungen werden bei der externen Daten- bzw. Dateiintegration die umfangreichsten Anforderungen gestellt, denn es sind strukturell unterschiedlichste Systeme in einer webbasierten Anwendung zusammenzuführen. Neben ERP- und beispielsweise Feinplanungssystem sind Daten aus einem CRM-System oder aus externen Datenstrukturen zu integrieren. Technologisch ergibt sich damit ein ähnliches Bild wie im Falle von VMI und ATP, nur dass die Anzahl der zu integrierenden Systeme deutlich höher ist. Neben der Integration der Daten ist darüberhinaus auch die inhaltliche Verknüpfung notwendig, z. B. zwischen Artikelnummern, Bilddaten und Rechnungsdaten.

Das Integrationsprinzip bei der webbasierten Integration mit LAMP ist auch bei Portallösungen der native Datenbankzugriff bzw. die Anbindung via ODBC auf die zu integrierenden datenhaltenden Systeme. Für diese Datenintegration werden die PHP-Funktionen `odbc_connect` oder `mysql_connect` bei MySQL-Datenbanken verwendet, für andere Datenbanken wie Informix oder Oracle existieren eigene PHP-Funktionen (`ifx_connect`, `oci_connect`, usw.), über die der Zugriff erfolgt.

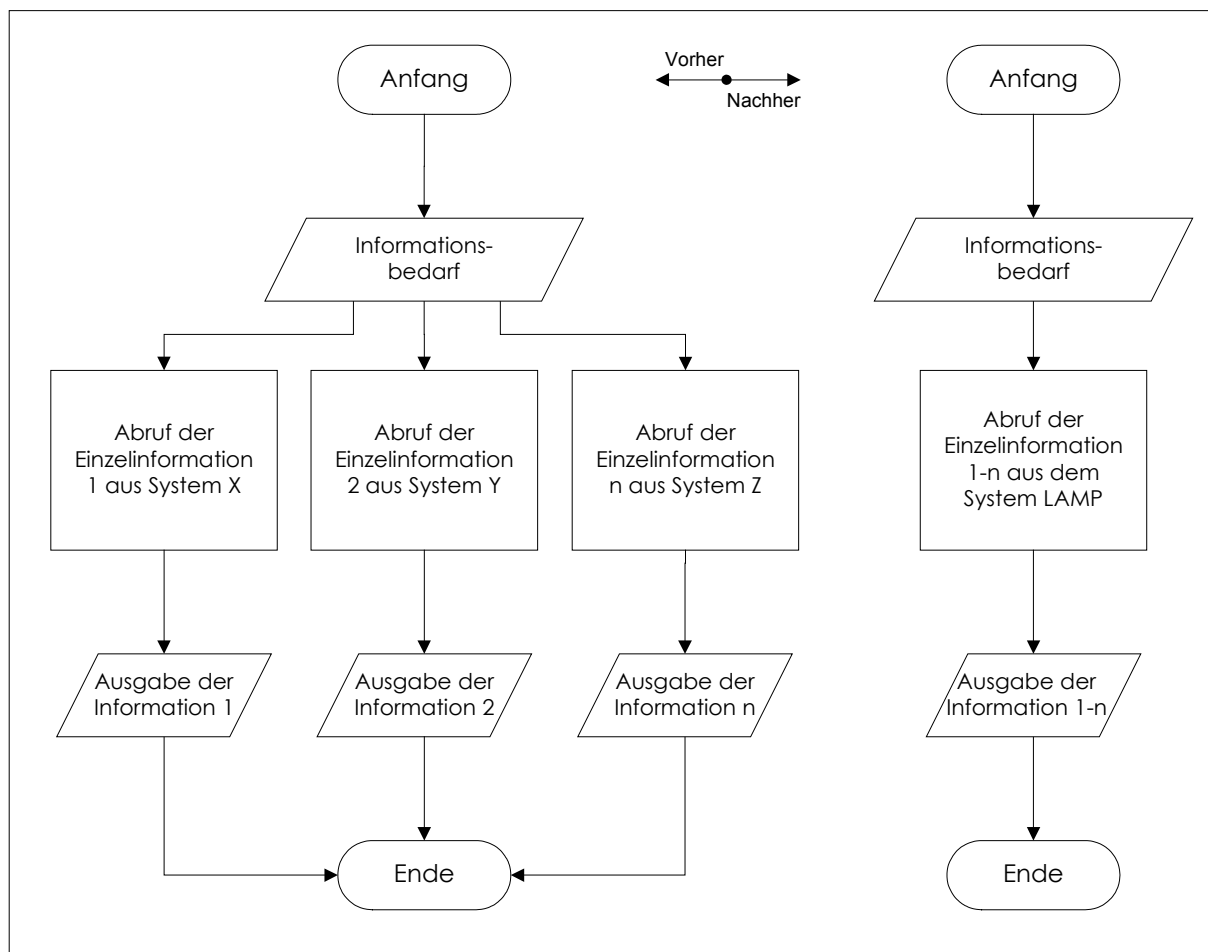


Abbildung 5.30: Prozessstruktur bei einer Portallösung

Die zusätzlich erforderliche Integration von Datenstrukturen aus Dateisystemen (z. B. Flat-Files oder Bilddateien) wird über spezielle PHP-Funktionen realisiert, die für die Unterstützung von Dateisystemen vorhanden sind, beispielsweise fopen für die Einbindung von Daten, die auf ftp-Servern vorhanden sind. Sofern erforderlich werden die aus den Datenquellen extrahierten und aufbereiteten Informationen in einer MySQL-Datenbank strukturiert abgelegt, im Rahmen eines Data-Warehouses inhaltlich miteinander verknüpft und schließlich über den Webserver zur Verfügung gestellt. Die hierbei zu berücksichtigenden Integrationselemente sind in der Abbildung 5.31 dargestellt.

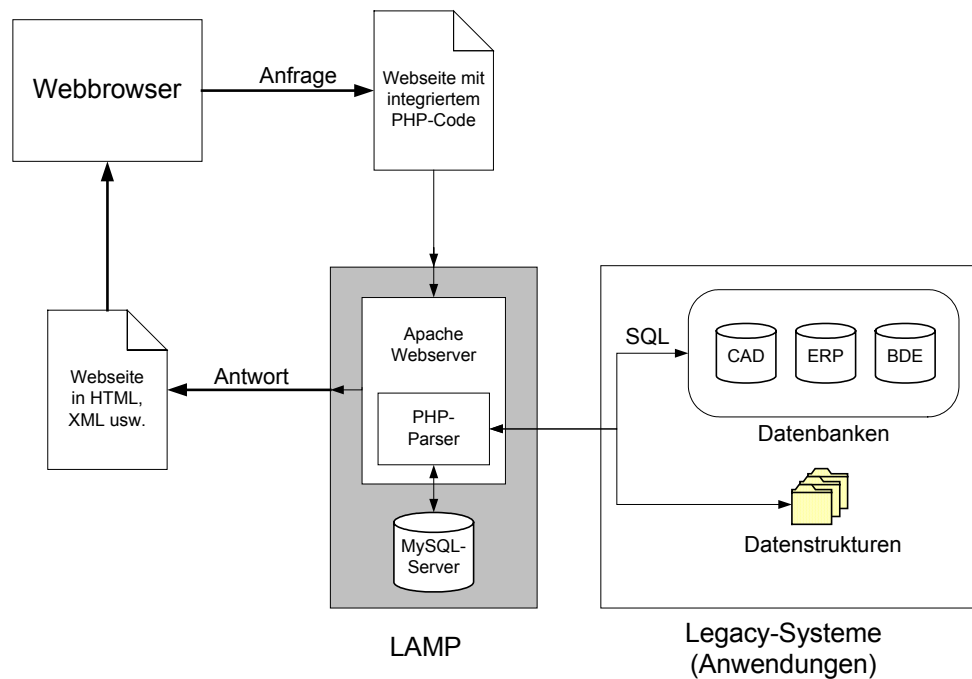


Abbildung 5.31: LAMP-Integrationselemente bei Portallösungen

Auf diese Art können praktisch alle unternehmensinternen Informationen, die sich in Datenbanken oder externen Datenstrukturen befinden und die innerhalb einer Supply Chain an die beteiligten Kunden und Lieferanten weitergereicht werden müssen, in einem integrierten Portal dargestellt und inhaltlich miteinander verknüpft werden.

5.5.4 Prozessdatenaustausch/EDI

Während bei der Umsetzung des VMI die Lagerbestandsinformationen eines Kunden über ein Anfrage-Antwort-Prinzip mittels Webbrowser zur Verfügung gestellt wurden, stellen Lieferketten mitunter weitreichendere Anforderungen an eine Echtzeitdatenverarbeitung, wie sie im Sinne des **RTE**¹⁴⁴ verstanden werden. Beim Real-Time Enterprise sind Kunden und beteiligte Partnerfirmen transparent eingebunden und tauschen Informationen über alle relevanten Ereignisse entlang der gemeinsamen Prozesse zeitnah aus [118].

Neben der Datenintegration, die im Wesentlichen den Lösungsansatz bei den bisher vorgestellten SCM-Prozessen darstellt, ist im Rahmen der Anwendungsintegration mit der Prozessintegration eine zweite bedeutsame Integrationsebene vorhanden. Durch die

¹⁴⁴ Real Time Enterprise

Prozessintegration erfolgt unter Verwendung von integrierten Anwendungen die Abbildung der Unternehmensprozesse innerhalb der internen und der externen Supply Chain. Während bei den internen Prozessen unternehmensinterne Anwendungen miteinander kommunizieren, sind es bei den externen Prozessen sowohl unternehmensinterne als auch -externe Anwendungen, für die Anwendungsintegration erforderlich ist. Diese Prozesse werden beispielsweise in Form von Workflowmechanismen abgebildet, bei denen zuvor definierte Teilprozesse aufeinander aufbauen und eine Weiterleitung von einem zum anderen Teilprozess nur in einer vorgegebenen Reihenfolge möglich ist.

Seit den achtziger Jahren wird für den Austausch von B2B-Prozessdaten in unternehmensübergreifenden Teilprozessen ein belegloses elektronisches Übertragungsverfahren für Auftrags-, Bestell- und Rechnungsdaten eingesetzt: EDI. Hierbei handelt es sich per Definition um einen Datenaustausch kaufmännischer Geschäftsdaten mittels Computer-Computer-Dialog in einem standardisierten Format [119]. EDI besteht einerseits aus dem Kommunikationssystem, das die DFÜ durch eine Point-to-Point-Verbindung ermöglicht, und dem Konvertierungssystem, das die Daten in standardisierte Nachrichtenformate wie das branchenübergreifende Datenaustauschformat EDIFACT oder das automobilbranchenspezifische Austauschformat ODETTE konvertiert. Zunehmend an Bedeutung gewinnt in den letzten Jahren Web-EDI, bei dem Daten beispielsweise in ein Web-Formular eingegeben und in normgerechte EDI-Nachrichten konvertiert und weitergeleitet werden. Hier setzt der webbasierte Prozessdatenaustausch mittels LAMP an.

Zur Ermittlung der für den elektronischen Datenaustausch mit Webtechnologien möglichen Integrationswege werden zunächst die im Rahmen von Webservices verwendeten Kommunikationsarten betrachtet. Grundsätzlich sind zwei Arten zu unterscheiden, nämlich die Kommunikation

- über URL via ftp, http und ssl oder
- über Nachrichten wie Email via SMTP und POP3.

Besondere Bedeutung hat in diesem Zusammenhang das Datenaustauschformat XML gewonnen. Die Extensible Markup Language gilt als Erweiterung der Seitenbeschreibungssprache HTML und ist ein vom **W3C**¹⁴⁵ definierter Standard für strukturierte Dokumente als Basis für Webseiten sowie für den Austausch von strukturierten Daten im Web. Im

¹⁴⁵ World Wide Web Consortium

Zusammenhang mit XML besitzen noch **XSDL**¹⁴⁶ (Schemabezeichnung für XML), WSDL (Schnittstellenbeschreibung von einzelnen Webservices) und SOAP (legt fest, wie Argumente an Web Services übergeben werden) eine besondere Bedeutung. Auf dieser Basis stellt XML eine weltweite branchenunabhängige Metasprache für das Definieren von Dokumenttypen dar. Sie geht jedoch über die Layoutbeschreibung von HTML hinaus, da sie neben Inhalt und Layout ergänzend die Struktur eines beliebigen Dokumenttyps (DTD, enthält die einzelnen möglichen Tags und Attribute für ein XML-Dokument) definieren kann [120].

Diese Ergänzung bietet dem elektronischen Datenaustausch eine völlig neue Perspektive im Bezug auf den kollaborativen Informationsfluss entlang der Supply Chain. Vergleichbar mit dem EDI-Datentransfer werden bei der prozessorientierten Übertragung von Daten zusätzliche Informationen transferiert, die den Empfängeranwendungen beschreiben, um welchen Datentyp es sich handelt und wie mit den übertragenen Daten verfahren werden soll. Die Anwendungsbereiche liegen insbesondere im Austausch von Datenbankinhalten, elektronischen Artikelkatalogen und ERP-Systemdaten. Am Beispiel eines Datenaustauschprozesses, der zwischen den ERP-Systemen zweier oder mehrerer Geschäftspartner abläuft, ist der Ablauf eines XML-gestützten Vorgangs schematisch in der Abbildung 5.32 dargestellt.

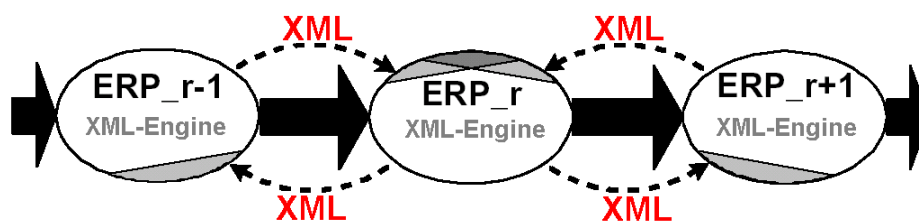


Abbildung 5.32: Kommunikation zwischen ERP-Anwendungen via XML

Grundsätzliche Voraussetzung für diesen Prozess ist zunächst, dass die beteiligten Partner ERP-Systeme besitzen, die XML-Nachrichten über XML-Engines verarbeiten können. Das erste ERP-System stellt gemäß vorgegebener XML-Schemadefinition einen XML-Datenstrom zur Verfügung. Die hierdurch adressierte XML-Engine des Zielsystems gibt diesen Datenstrom an das ERP-System weiter. Hier erfolgt eine Verarbeitung der empfangenen Daten und anschließend die Weiterleitung der Ergebnisse an das nächste System oder zurück zum Ausgangssystem, ebenfalls über die XML-Schnittstelle. Im nächsten ERP-System wiederholt sich dieser Vorgang, und er endet erst dann, wenn die

¹⁴⁶ eXtensible Style Definition Language

Supply Chain bzw. der Supply Chain-Prozess vollständig abgearbeitet ist. Sofern zwei Unternehmen in der Lage sind, XML-Nachrichten über XML-Engines verarbeiten zu können und die benötigten Daten über die XML-Schemadefinition aufeinander abgeglichen sind, ist der Informationsfluss über die XML-Engine des einen ERP-Systems zur Engine des nächsten Systems in Form von automatisch generierten XML-Datenströmen quasi in Echtzeit möglich. Dadurch können Geschäftsprozesse unternehmensübergreifend gesteuert sowie Medienbrüche vermieden werden. Zeit und Kosten werden gespart, weil die Arbeitsschritte und Systemeingaben durch die automatische Verarbeitung der XML-Daten minimiert und die Dokumente in Internetgeschwindigkeit versendet werden.

Die XML-Schnittstellen spielen in dem beschriebenen Prozess eine zentrale Rolle. Sofern die eingesetzten ERP-System nicht über XML-Engines verfügen, ist der Datenaustausch via XML nicht möglich und die Kommunikation unterbrochen. Um dies zu vermeiden, sind nicht XML-fähige Anwendungen durch Anwendungsintegration XML-fähig zu machen. Dies geschieht unter Einsatz der LAMP-Umgebung, die die Funktion einer XML-Engine wahrnimmt und die Anwendungssysteme auf diese Weise integriert. Zentrales Integrationselement ist dabei PHP. Die von einem entfernten System übermittelten XML-Daten werden zunächst dem PHP-Parser übergeben, der den als PHP-Funktion vorhandenen XML-Parser aufruft. Hier werden die XML-Daten analysiert, verarbeitet und in die Datenbank des ERP-Systems übertragen, indem die ERP-Datenbank skriptgesteuert durch nativen Zugriff oder via ODBC über ein PHP-Skript aktualisiert wird. Dadurch wird das ERP-System in den XML-Datenaustausch eingebunden und XML-fähig (vgl. Abbildung 5.33).

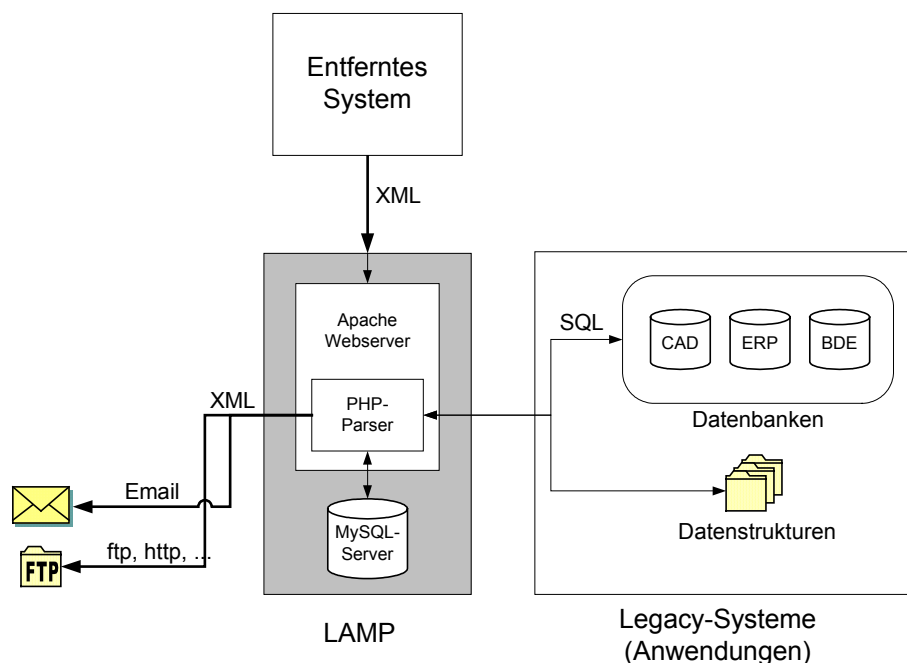


Abbildung 5.33: LAMP-Integrationselemente bei der XML-Kommunikation

In umgekehrter Reihenfolge erfolgt die Bereitstellung der ERP-Daten als XML-Datenstrom, der schließlich via Email, ftp, http usw. als XML-Nachricht zur Verfügung gestellt bzw. an das nächste System übertragen werden kann. Gleichermaßen können auch Datenbankinformationen und Datenstrukturen aus weiteren Systemen (BDE, CAD) als XML-Nachricht aufbereitet und an Folgesysteme gesendet werden.

Für das einfache Parsen, d. h. Einlesen und Verarbeiten von XML-Dokumenten, bietet sich die Verwendung des in PHP 4 integrierten XML-Parsers expat an [121]. Sofern dessen Event-basierte Verarbeitung nicht ausreicht, stellt die **DOM**¹⁴⁷-XML-Erweiterung eine Baumrepräsentation der Daten zur Verfügung. Die Funktionen der DOM-XML-Erweiterung basieren auf libxml, einer Bibliothek die unter anderem im Gnome-Projekt zum Einsatz kommt. In PHP 5 erfolgt die integrierte XML-Unterstützung durch die aus dem Gnome-Projekt stammende Bibliothek libxml2 [122]. Mit diesen Funktionen ist es auch möglich, XML-Strukturen einfach zu erzeugen.

Neben den bereits genannten Funktionen zum Erstellen und Parsen von XML bietet PHP 4 dank der Integration von Sablotron¹⁴⁸ auch Funktionen zur Transformation von Dokumenten aus einem XML-Format in ein anderes entsprechend der Transformationssprache für XML, beispielsweise über **XSL**¹⁴⁹-Transformationen (XSLT). Bei XSLT handelt es sich um eine vom W3C entwickelte Sprache für die Konvertierung von XML-Dat(ei)en in ein anderes Format. So werden XML-Datenströme je nach Integrationsanforderung konvertiert und die korrekte Verarbeitung der Daten im Folgesystem sichergestellt. Innerhalb der internen Supply Chain stellt ein solches Folgesystem eine zu integrierende IT-Anwendung dar, bei der externen Supply Chain handelt es sich beim Folgesystem um ein Anwendungssystem, das bei einem Kunden, Lieferanten oder Geschäftspartner vorhanden ist und die Prozessdaten in einem vorgegebenen Format zur Verfügung gestellt bekommen muss. Die LAMP-Umgebung stellt in beiden Fällen die Grundlagen zur Verfügung, einerseits XML-Datenströme direkt zu verarbeiten und andererseits die am XML-Datenaustauschprozess beteiligten Systeme im Rahmen einer Anwendungsintegration zusammenzuführen.

Die Möglichkeiten von LAMP gehen aber noch über die einer XML-Engine hinaus, denn im Rahmen von webbasierten Workflowmechanismen ist mit LAMP die Abbildung ganzer Prozessstrukturen darstellbar, sofern das Prozessmodell und die Anforderungen daraus vorliegen. Neben der internen Supply Chain bietet sich dabei ebenfalls der Einsatz in der

¹⁴⁷ Document Object Model

¹⁴⁸ XSL-Parser mit XSLT 1.0-Unterstützung

¹⁴⁹ eXtensible Stylesheet Language

externen Supply Chain an, so dass Lieferanten und Kunden derart einbezogen werden können, dass sie beispielsweise über einen einfachen Webbrowser als Frontend den ihnen zugeordneten Prozessschritt als Teil des Gesamtprozesses bearbeiten. Damit wechselt die Perspektive vom reinen Datenaustausch hin zur Prozesskommunikation, innerhalb derer Daten ausgetauscht werden müssen.

6 Realisierung

Unter Verwendung der LAMP-Umgebung wurden im vorangegangenen Kapitel automotiv-typische SCM-Prozesse dargestellt und die erforderlichen Technologien beschrieben. Auf den folgenden Seiten wird nun die Umsetzung dieser Konzepte in einer mittelständischen Unternehmens- und IT-Umgebung dargestellt. Die Ergebnisse werden abschließend diskutiert.

Die Betrachtung der Prozesse erfolgt aus der Sicht der Anwendungsintegration unter Verwendung des Internets als Kommunikationsmedium. Auf die für die Internetkommunikation erforderlichen Sicherheitsarchitekturen wird dabei vor dem Hintergrund der Zielsetzung dieser Arbeit nicht eingegangen, hier sei vielmehr auf entsprechende Literatur verwiesen [123, 124, 125].

6.1 Wahl des Unternehmens

Die Realisierung des Konzeptes wird vorgenommen in der IT-Umgebung eines mittelständischen Automobilzulieferers, der auf dem Gebiet der Kunststoffverarbeitung tätig ist. An vier Standorten werden dort bei einem Gesamtumsatz von 60 Mio. EUR/a mit insgesamt 500 Mitarbeitern technisch anspruchsvolle Kunststoffteile im Spritzgussverfahren für die Automobilindustrie hergestellt, die den Produktbereichen Lüfter und Luftführungsteile, Lichttechnik, Filter und Innenraum zuzuordnen sind. Über die Teileproduktion hinaus werden kundenspezifische Entwicklungsaufgaben und Dienstleistungen auf den Gebieten der Konstruktion und des Rapid Prototyping wahrgenommen, die Belieferung der Kunden erfolgt bei einigen Produktlinien JIT bzw. JIS. Von den vier Produktionsstandorten befinden sich drei in Deutschland (Hessen, Sachsen, Baden-Württemberg), wobei am Standort in Hessen zentrale Verwaltungsfunktionen angesiedelt sind. Mit dem vierten Standort, der sich in Mexiko befindet, wird die internationale Ausrichtung des Unternehmens dokumentiert. Aufgrund der Größe und der Struktur ist das Unternehmen dem Mittelstand zuzuordnen. Charakteristisches Merkmal des Unternehmens ist die heterogene IT-Systemstruktur, die zur Abbildung von internen und externen Supply Chain Management-Prozessen die beschriebenen Integrationswege über LAMP erfordert.

6.2 Software- und Systemumgebung

Wie in den mittelständischen IT-Systemumgebungen üblich, ist die IT-Struktur des betrachteten Unternehmens heterogen aufgebaut. Neben einem zentralen ERP-System (XPPS des Herstellers Agilisys (ehemals BRAIN International AG)) sind eine Reihe von weiteren Systemen im Einsatz bzw. in Planung (CAQ-, Mail-, Intranet-, Feinplanungssystem usw.), diese sind in der Abbildung 6.1 dargestellt.

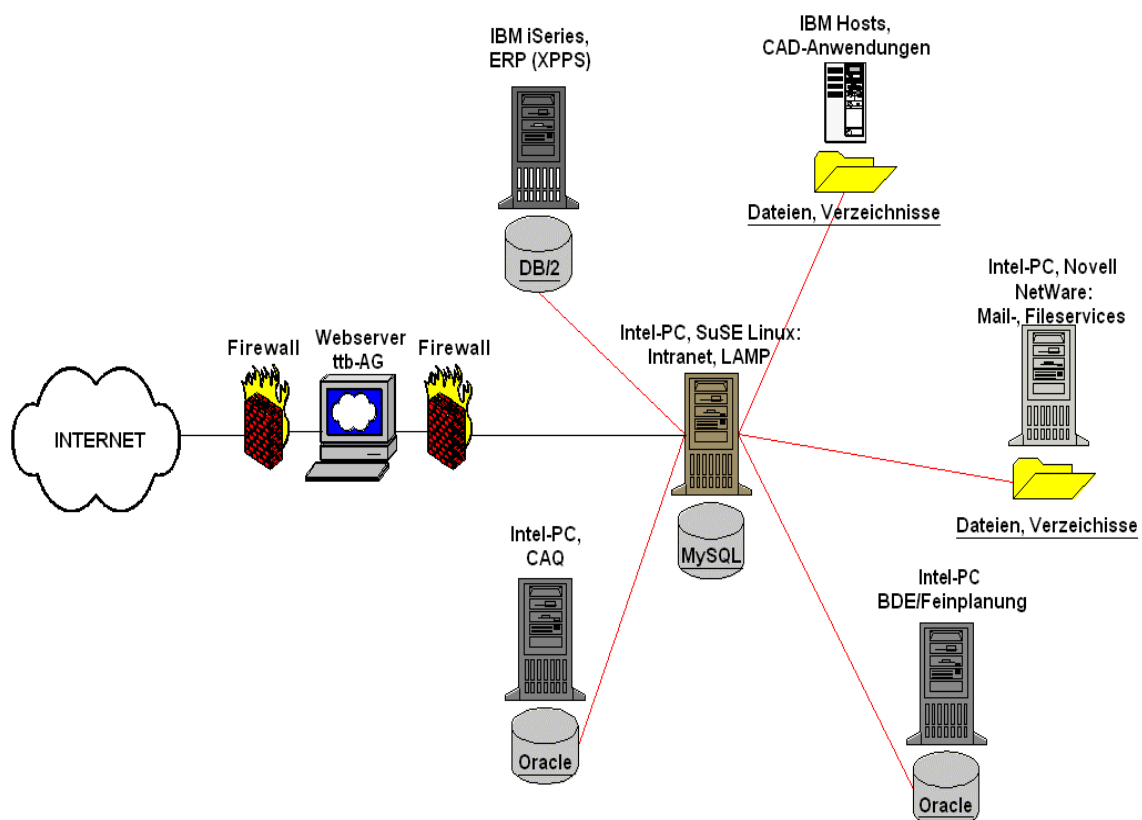


Abbildung 6.1: Software- und Systemumgebung des betrachteten Unternehmens

Die Systeme basieren zumeist auf Datenbanken wie DB/2, Oracle und MySQL. Darüberhinaus sind Datenquellen bzw. -strukturen vorhanden, die in Form von Dateien und Verzeichnissen vorliegen. Alle Datenquellen werden über einen zentralen LAMP-Integrationsserver zusammengeführt, der über eine durch Firewallimplementierungen geschützte Verbindung in der Lage ist, über das Internet oder intern die angeforderten Daten und Informationen über einen Webserver zur Verfügung zu stellen.

Die LAMP-Systemumgebung besteht gemäß Tabelle 4 aus einer Linux-Installation, einem Apache-Webserver, einer MySQL-Datenbank sowie einem PHP-Parser. Die für die Realisierung verwendeten Systeme sind in den folgenden Releaseständen vorhanden:

LAMP	Releasestand
Linux	SuSE Linux 8.0, Kernel 2.4.18, interne IP-Adresse : 192.168.1.222
Apache	Apache Webserver 1.3.7 mit konfiguriertem Modul mod_php4
MySQL	MySQL 3.23.48
PHP	PHP 4.3.0, kompiliert mit UNIX-ODBC-Unterstützung

Tabelle 4: LAMP-Umgebung in dem Pilot-Unternehmen

Zur Installation der einzelnen LAMP-Module wird hier auf die in den Modulen enthaltenen Installationsbeschreibungen sowie auf weiterführende Literatur verwiesen [104, 126]. Auf der Seite der Anwendungssysteme besteht die Infrastruktur gemäß Tabelle 5 aus

Systeme	Installation
ERP	IBM iSeries mit OS/400¹⁵⁰ V5R2 Agilisys (ehemals BRAIN International AG), XPPS 2.20
BDE	Compaq ML 350 mit MS Windows 2000, SP 2 Barco, BARCOCIM BDE/Leitstand
CAQ	Compaq ML 310 mit MS Windows 2000 Server, SP 2 IBS, QSYS 6.04
Mailsystem	Novell NetWare 4.11, Service Pack 9 (gleichzeitig Filesystem) Tobit, David Professional 6.6
CAD	IBM AIX IBM Catia

Tabelle 5: Installierte Software-Systeme in dem Pilot-Unternehmen

Eingesetzt werden neben den hier genannten Server-Systemen von IBM und Compaq auch IBM-kompatible Rechner von OEM-Herstellern. Als Zugang zum Internet steht eine synchrone Datenleitung mit einer Übertragungsrate von 2 Mbit/ s zur Verfügung. Über diesen Zugang ist der LAMP-Server im Internet verfügbar.

¹⁵⁰ Operating System /400 (IBM)

6.3 Programmtechnische Umsetzung

Unter Verwendung der LAMP-Umgebung werden im Folgenden die hier nachfolgend genannten SCM-Prozesse bzw. -Konzepte umgesetzt:

- VMI,
- ATP,
- die Portalanbindung und
- der Prozessdatenaustausch / EDI.

Diese Prozesse entsprechen auch den Detailkonzepten aus Kapitel 5. Die Wahl dieser Beispielprozesse bedeutet nicht, dass das Konzept nur auf diese Prozesse ausgelegt ist. Diese Prozesse sind deshalb gewählt worden, da ihr Inhalt bereits schon einmal beschrieben worden ist.

6.3.1 VMI

Das VMI ermöglicht die Verwaltung eines Lagers durch den Lieferanten, indem ihm die betreffenden Informationen bzgl. des Lagerbestandes zur Verfügung gestellt werden. Hierfür sind die Bestandsdaten aus dem Lagerwirtschaftssystem an einen Internet-Client des Lieferanten zu übermitteln, insofern ist eine Verbindung zwischen beiden Anwendungen erforderlich. Gemäß Abbildung 6.1 ist das Lagerwirtschaftssystem über das ERP-System „XPPS“ realisiert. Die zugrundeliegende Hardware ist eine IBM iSeries-Plattform (ehemals AS/400), auf der in DB2-Datenbanken die Lagerbestandsdaten abgelegt sind.

Der Integrationsweg verläuft über einen ODBC-Zugriff auf die DB2-Datenbanken mittels PHP, die auf diese Weise extrahierten Daten werden schließlich über den Apache Webserver dem Webbrowser zur Verfügung gestellt. Zunächst sind die Zugriffsberechtigungen auf die Datenbanken sowie die Anmeldeinformationen auf der iSeries zu konfigurieren. Dies ist für den ODBC-Zugriff erforderlich und in der Datei `odbc.ini` (vgl. Abbildung 6.2) in der PHP-Distribution beschrieben. Neben den Anmeldeinformationen für das iSeries-System sind darin weitere systemspezifische Parameter, die für die Datenübertragung erforderlich sind, angegeben (z. B. Default-Datenbank und Datenkompression).

Parameter	Wert	Parameter	Wert
Description	ISeries Access ODBC Driver	DefaultPkgLibrary	BIDBD220
Driver	ISeries Access ODBC Driver	DefaultPackage	A/DEFAULT(IBM),2,0,1,0,512
System	192.168.11.12	AllowDataCompression	1
UserID	QUSER	LibraryView	0
Password	QUSER	AllowUnsupportedChar	0
Naming	0	Trace	0
DefaultLibraries	BIDBD220	Prefetch	1
ConnectionType	3	BLOCKFETCH	1
CommitMode	2	BLOCKSIZE	128
ExtendedDynamic	1	CCSID	819

Abbildung 6.2: Konfigurationsdatei odbc.ini

Als Frontend für den Lieferanten dient ein Webbrowser. Ausgangspunkt ist die Startseite, über die die Anmeldung eines Lieferanten erfolgt, der die Lagerbestände seines Kunden überprüfen möchte (Abbildung 6.3).

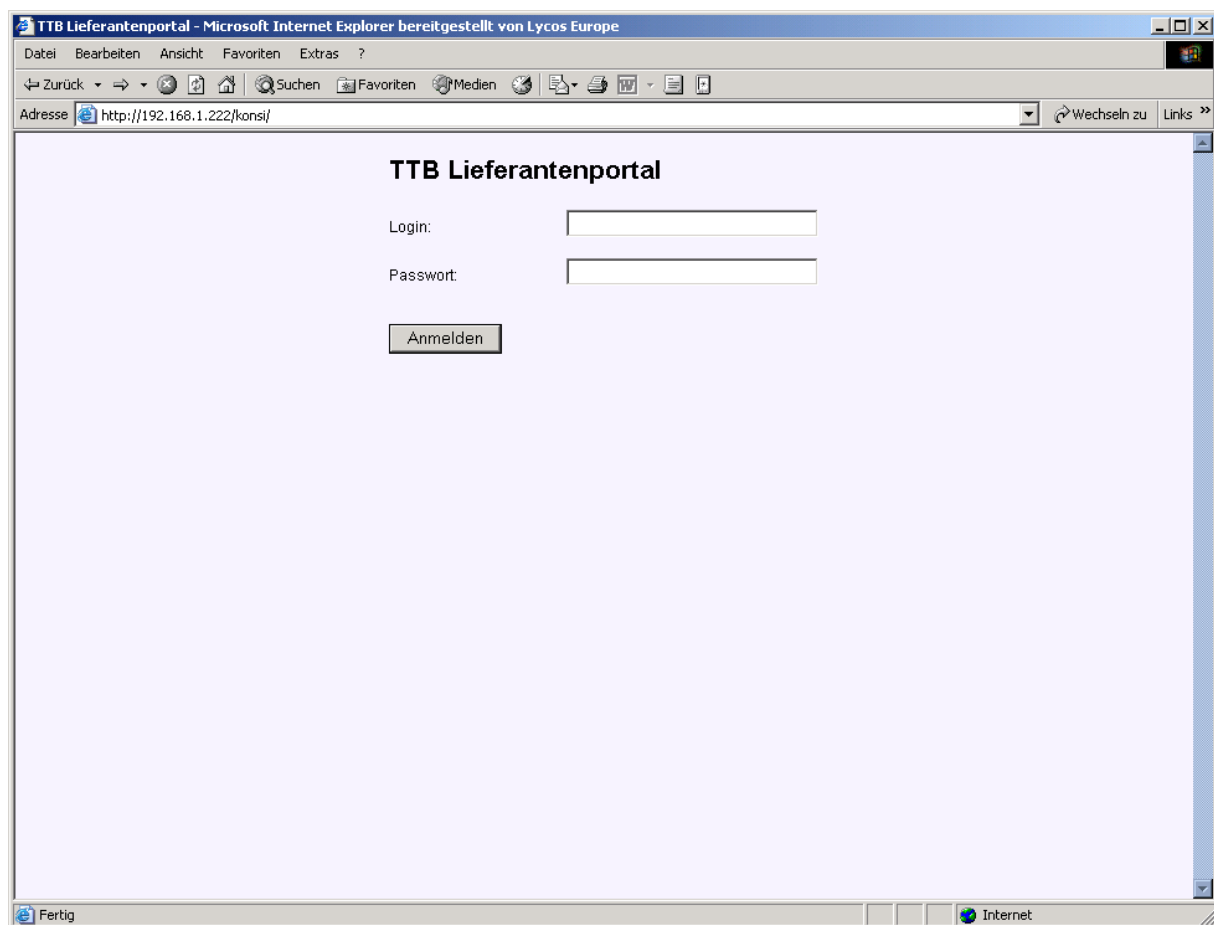


Abbildung 6.3: Loginseite eines Lieferanten für das VMI

Hierin wird der Lieferant zunächst zur Eingabe seiner Accountinformationen (Login bzw. Lieferantenummer mit Passwort) aufgefordert, die gegen die Einträge einer MySQL-Datenbank geprüft werden. Bei erfolgreichem Login ist der Lieferant dem System bekannt, bei erfolglosen Login-Versuchen wird eine Fehlermeldung ausgegeben und der Zugang zu den Bestandsdaten verweigert.

Für die Umsetzung der Berechtigungsstruktur, die für die Lieferantenanmeldung erforderlich ist, werden innerhalb einer MySQL-Datenbank (konsi) zwei Datenbanktabellen angelegt: zum einen die Datenbanktabelle auth (Tabelle 6), die die Informationen über den Lieferanten beinhaltet, nämlich Loginname (uname), Lieferantename (firm_name) sowie ein **MD5**¹⁵¹-verschlüsseltes Passwort (pword), und zum anderen die Datenbanktabelle rights (Tabelle 7), in der die Rechtestruktur des Lieferanten definiert ist (User-ID (user), Berechtigung für das Werk (werk) und Berechtigung für das Lieferantenlager (lager)). Die zu den Tabellen gehörenden Datenbankbeschreibungen sind im Anhang 12.1 (Abbildung 12.1 und Abbildung 12.2) erläutert.

id	uname	firm_name	pword
1	test	Horna	098f6bcd4621d373cade4e832627b4f6
2	tff	tff	3188a8301b98ccdc7d297e902041bd68

Tabelle 6: Tabelle auth der Datenbank konsi

user	werk	lager
1	100	K1
2	100	K1

Tabelle 7: Tabelle rights der Datenbank konsi

Sind nach erfolgreichem Login keine Teilenummern im Lager vorhanden, die lieferantenbezogen angezeigt werden können, erfolgt eine Fehlermeldung, ansonsten werden die Teilenummern mit Bestandsdaten ausgegeben, die im VMI des Lieferanten hinterlegt sind.

¹⁵¹ Message Digest Algorithm 5

Zum Extrahieren dieser Lagerbestandsdaten aus der DB2-Datenbank des ERP-Systems (vgl. Abbildung 6.4) dient zunächst die PHP-Funktion `odbc_connect`. Nach dem Herstellen der Verbindung zur Datenbank werden aus den Datentabellen LGBS und TEIL die Inhalte der Felder Teilenummer (lstenr), Lagerbestand (lslgbe) sowie Teilebezeichnung (tebez1 und tebez2) über ein SELECT-Kommando via SQL ausgelesen. Im SQL-Code werden dazu die Tabellen TEIL und LGBS verwendet, die die für den Lieferanten zugänglichen Teilenummern aus der DB2-Datenbank selektieren.

Die für die Login-Prozedur erforderliche Verbindung zur MySQL-Datenbank konsi wird durch die PHP-Funktion `mysql_connect` hergestellt. Die beiden Datenbanktabellen `auth` und `rights` sind dabei über die Felder `id` und `user` verbunden. Zwischen den Tabellen LGBS und `rights` erfolgt die Zuordnung der Werksnummern, die ebenfalls zwischen den Tabellen TEIL und `rights` vorhanden ist. Zur Anzeige des für den Lieferanten zugänglichen Lagers werden die Lagernummern in den Tabellen LGBS und `rights` selektiert.

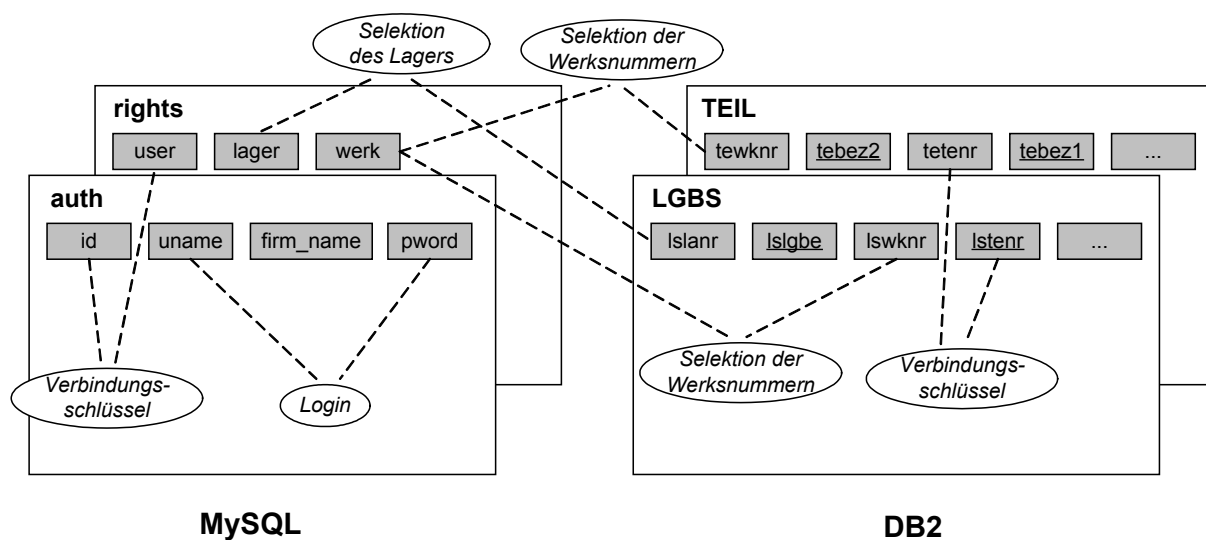


Abbildung 6.4: Datenbanktabellen der Datenbanken MySQL und DB2 für das VMI

Die auf diese Weise ausgewählten Datensätze aus den Datenbanktabellen werden an den Browser des Lieferanten gesendet, was in der Abbildung 6.5 dargestellt ist.

TTB Lieferantenportal [Logout](#)

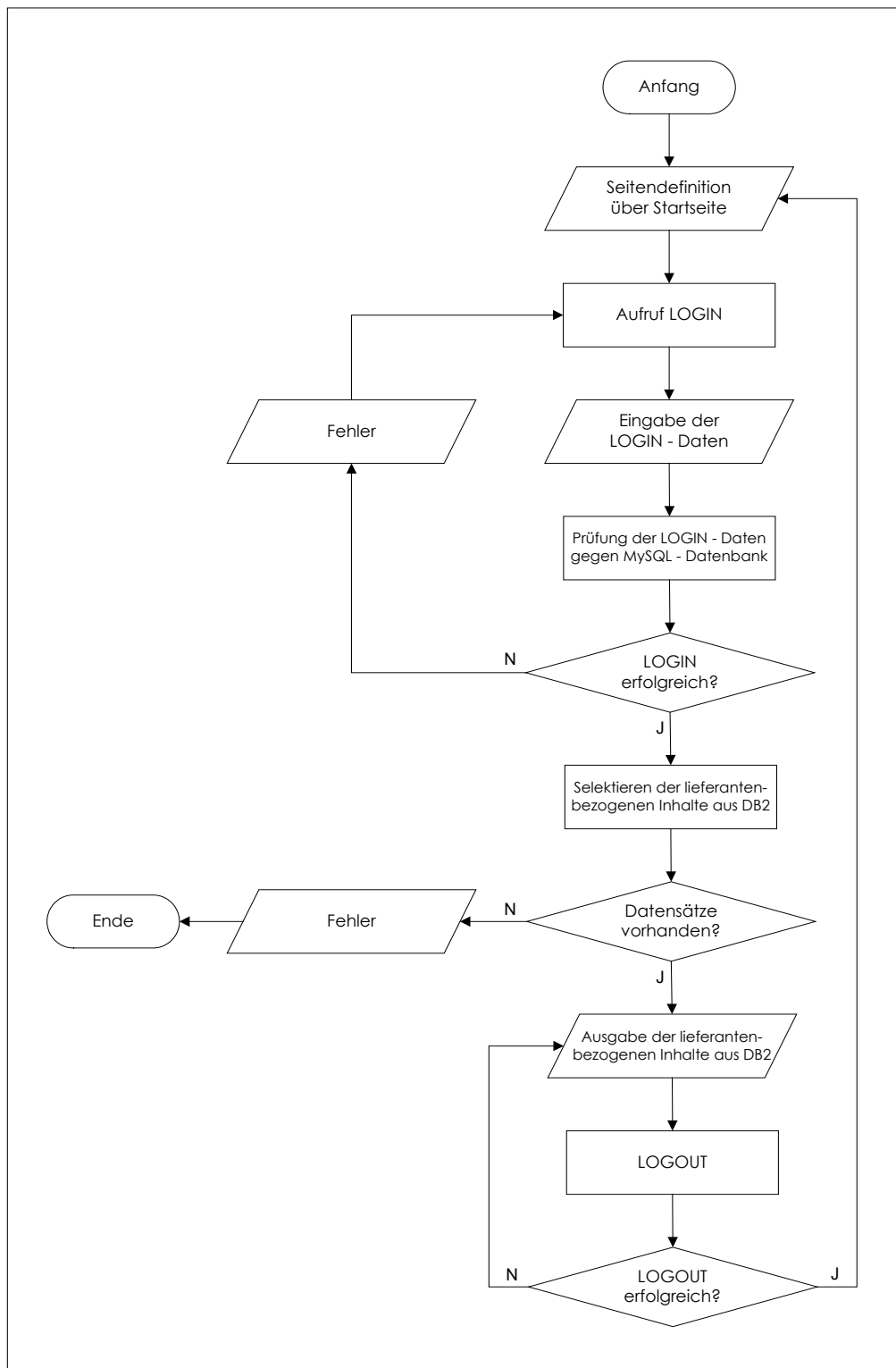
Werk: **100**

TTF-Artnr	Bestand	Bezeichnung 1	Bezeichnung 2
000169	600,00	Karton V 608900	545 x 465 x 450 mm IMC 110
000435	2 000,00	Begletpapiertaschen LDPE	240 x 160 + 25 mm, C5
000778	0,00	Polybeutel	400 mm x 600 mm, 50 my
000953	0,00	Punkt rot	Durchmesser 50 mm
207004	6 900,00	Umreifungsband PT13 PB	13 mm, 0,6 mm, 2300 mtr.
207009	5 000,00	Klammer für Umreifungsband	
301005	214,00	Karton 370 x 370 x 350 mm	für 300.51/1/2/3/4/5
301014	0,00	Faltschachtel A3 LL Votex	101.11 VO Filterset
301015	0,00	Faltschachtel B3 LL Votex	101.22 VO Filterset
301029	110,00	Karton 1170 x 770 x 750 mm	3-wellig Q.2.92
302003	0,00	Zuschnitt	1500 x 40 mm, Q. 1.10
302010	2 100,00	Wellpapp-Zuschnitt plano	780 x 1180 mm, Q. 1.30
302013	400,00	Wellpapp-Zuschnitt	925 x 1210 mm, Q. 1.1
302020	2 200,00	Wellpapp-Zuschnitt	1100 x 900 mm, Q. 1.0
302025	1 710,00	Wellpapp-Zuschnitt plano	1193 x 250 mm Q. 2.7
302026	1 500,00	Quersteg mittig geschlitzt	790 x 400 mm Q. 2.7
302027	1 200,00	Wellpapp-6er-Gefache	1193x790x350 mm Q 1.01
302035	0,00	Schlitzverpackung flach	365x365x61 mm, Q.1.02
302039	430,00	Wellp.Gefach m. Blindfach	1180x390 + Blindfach 80 mm
302040	550,00	Wellpapp-Ring Durchm.390	
302041	0,00	J-Profil 1180 x 780 mm	Q. 1.02
302042	0,00	Wellpapp-Zuschnitt Q. 1.01	1170x70/70x760 (Gefache)
302046	860,00	Abtrennung 1630 x 250 mm	passend zu 302041
307002	240,00	Einwegpalette 1200 x 800	behandelt USAMex. 250 kg
307004	0,00	Einwegpalette 974x1140 mm	
307006	50,00	Einwegpalette 1200 x 800	behandelt USAMex. 900 kg

Abbildung 6.5: Lagerbestandsinformationen aus dem ERP-System im Browserfenster des Lieferanten

Hier sind die zu den einzelnen Artikelnummern gehörenden Lagerbestände sowie die Bezeichnungen der Artikel in tabellarischer Form werksbezogen ausgegeben, wodurch der Lieferant die Informationen darüber erhält, welche Teile in welchen Mengen im Lager seines Kunden vorhanden sind. Die primäre Anforderung an ein VMI, nämlich die Bereitstellung der Informationen über diese Lagerbestände, ist damit bereits erfüllt. Über eine Logout-Möglichkeit kann der Lieferant seine Abfrage und damit das Programm beenden, das in diesem Fall wieder auf die Startseite verzweigt.

Für diese Umsetzung ist ein einziges PHP-Skript ausreichend, das im Anhang 12.2 (Abbildung 12.3 - Abbildung 12.17) der Übersichtlichkeit halber ausführlich in Form von sechs Teilskripten wiedergegeben und erläutert ist. Stattdessen ist hier in der Abbildung 6.6 das Ablaufdiagramm dargestellt, welches die oben beschriebenen Funktionalitäten beschreibt.

**Abbildung 6.6: Ablaufdiagramm zum VMI**

Es ist leicht nachvollziehbar, dass sich auf diese Weise prinzipiell alle im Kundensystem darstellbaren Informationen in das Frontend des Lieferanten in Echtzeit übertragen lassen. Ergänzend sind hier beispielsweise noch die bisherigen Wareneingänge auf der Seite des Kunden bzw. die verbuchten Lieferungen anzuzeigen, damit der Lieferant einen Überblick

darüber erhält, ob beispielsweise eine Lieferung auf dem Weg zum Kunden, allerdings noch nicht im Kundensystem verbucht ist, und er dadurch eine Doppellieferung vermeiden kann. Gleiches gilt für die Definition von Minimal- und Maximalbeständen oder für eine automatisierte Benachrichtigung bei Unter- oder Überschreitung dieser Bestandsgrenzen. Das Zugriffsprinzip ist hier ebenfalls der Datenbankzugriff via ODBC auf die iSeries-Datenbank, die in diesem Fall die Informationen über die verbuchten Wareneingänge enthält.

6.3.2 ATP

Von externen Beschaffungsquellen abgesehen, können kundenseitige Bestellungen zu einem vorgegebenem Zeitpunkt entweder durch Lagerbestände oder durch die laufende Fertigung der benötigten Artikel gedeckt werden. Zusätzlich zur Information über die Lagerbestände sind bei ATP-Anfragen daher auch Informationen über Fertigungsaufträge zu berücksichtigen. Hierbei spielen u. a. die Endtermine von geplanten und laufenden Fertigungsaufträgen eine Rolle.

Während die Vorgehensweise zur Darstellung der Informationen über den Lagerbestand bereits im VMI beschrieben wurde, stellt sich hier die Frage, wie die Informationen über die Fertigungsaufträge einem Kunden zur Verfügung gestellt werden können. Diese Informationen sind in der vorhandenen Systemumgebung nur teilweise im ERP-System zu finden. Zwar findet dort die Einlastung der Fertigungsaufträge statt, die terminliche Detailplanung der einzelnen Aufträge wird jedoch in einem anderen System durchgeführt, dem Feinplanungssystem (BARCOCIM BDE / Leitstand). Das Feinplanungssystem arbeitet intern mit einer eigenen Datenbank von Oracle. Zur Darstellung der Feinplanungsinformationen ist daher ein Zugriff auf die entsprechende Oracle-Datenbank erforderlich.

PHP bietet mit den oci-Funktionen für den Datenbank-Connect zu Oracle eine native Datenbankunterstützung. Um beispielsweise die Artikelinformationen aus der Oracle-Datenbank des Feinplanungssystems auszugeben, dient folgendes Ablaufdiagramm (Abbildung 6.7):

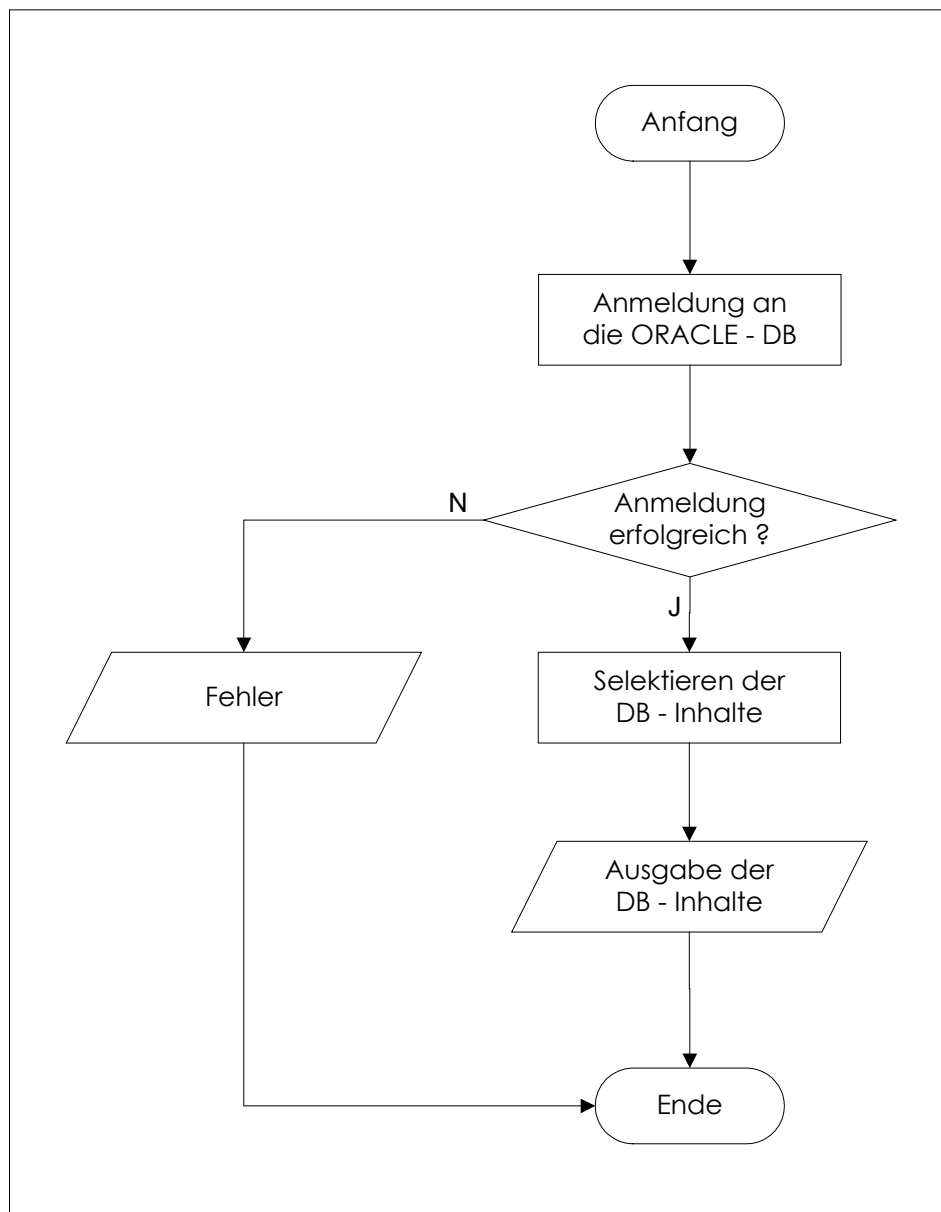


Abbildung 6.7: Ablaufdiagramm für die artikelspezifischen Inhalte einer Oracle-Datenbank

Darin erfolgt über die Anmeldung an der Oracle-Datenbank die Selektion und anschließend die Ausgabe der betreffenden Datenbankinhalte, bei fehlerhaftem Login-Versuch wird eine entsprechende Meldung ausgegeben. Das dieser Funktion zugehörige Programmskript ist im Anhang 12.3 (Abbildung 12.18 - Abbildung 12.20) ausführlich beschrieben.

Analog wie im Falle des VMI stehen damit die Daten der Oracle-Anwendung zur weiteren Verwendung im Webbrowser zur Verfügung. In einem letzten Schritt, auf den hier nicht weiter eingegangen wird, sind nun noch die beiden Datenpools, ERP-Daten aus der DB2-Datenbank einerseits und Feinplanungsdaten aus der Oracle-Datenbank andererseits, nach anforderungsabhängigen Kriterien mittels SQL-Syntax miteinander zu verbinden (zu joinen). Damit liegen innerhalb einer Integrationsoberfläche die gemeinsamen Daten über die Lager-

bestände und die Fertigungsaufträge vor, die innerhalb des im Rahmen einer ATP-Anfrage benötigten Datenbestandes erforderlich sind.

6.3.3 Portallösung

Aufbauend auf den informationstechnischen Integrationspfaden aus den bisher betrachteten Prozessen VMI und ATP ist die Anzahl der zu integrierenden Systeme bei einer Portallösung üblicherweise größer als zwei. Neben Bestands- und Fertigungsauftragsinformationen aus ERP- und Feinplanungssystem sind weitere Systeme wie Artikelkataloge, Mailsystem und CAQ-Lösungen zu integrieren (siehe Abbildung 6.1). Die Ausgangssysteme können hierbei sowohl Datenbanken als auch einzelne Dateien (flat files) beinhalten, die in einer Integrationslösung unter einer gemeinsamen Oberfläche abgebildet werden müssen. Insofern stellt eine Portallösung eine funktionale Integrationserweiterung der bisher beschriebenen Prozesse dar.

Die Einbindung von Datenbankstrukturen in die LAMP-Umgebung wurde bereits auf den vorangegangenen Seiten anhand der beiden SCM-Prozesse VMI und ATP beschrieben. Dabei wurde die Datenbankunterstützung von PHP für DB2- und Oracle-Datenbanken verwendet. Für eine Reihe von weiteren Datenbankformaten, die nativ von PHP unterstützt werden, findet die Integration in analoger Weise statt, daher wird im folgenden die programmtechnische Umsetzung für die Integration von Dateien, die innerhalb einer Dateistruktur zur Verfügung stehen, betrachtet (flat files). Im konkreten Fall handelt es sich um eine Dateistruktur innerhalb einer AIX-Umgebung von IBM.

In PHP stehen für die Integration von Dateisystemen eigene Funktionen zur Verfügung. Mit der Funktion `fopen` werden Dateistrukturen eingebunden und in einem Browserfenster dargestellt, die beispielsweise auf einem ftp-Server vorhanden sind. Die konkrete Umsetzung für einen Artikelkatalog, der die Artikelstammdaten via ODBC aus dem ERP-System (XPPS) bezieht und die Bilddateien zu den Artikeln aus einem entfernten System über einen ftp-Stream bereitgestellt bekommt, zeigt die nachfolgende Abbildung 6.8. Darin sind neben den Artikelinformationen aus dem ERP-System (Artikelnummer, Bestand und Bezeichnung) auch die zu den Artikeln gehörenden Bilddateien integriert.

TTB Lieferantenportal [Logout](#)

Werk: 100

Bild	TTF-ArtNr	Bestand	Bezeichnung 1	Bezeichnung 2
	000169	360,00	Karton V 608900	545 x 465 x 450 mm IMC 110
	000435	2 000,00	Begleitpapiertaschen LDPE	240 x 160 + 25 mm, C5
	000778	0,00	Polybeutel	400 mm x 600 mm, 50 my
	000953	0,00	Punkt rot	Durchmesser 50 mm
	207004	4 600,00	Umreifungsband PT13 PB	13 mm, 0,6 mm, 2300 mtr.
	207009	3 000,00	Klammer für Umreifungsband	
	301005	200,00	Karton 370 x 370 x 350 mm	für 300.51/1/2/3/4/5
	301014	0,00	Faltschachtel A3 LL Votex	101.11 VO Filterset
	301015	0,00	Faltschachtel B3 LL Votex	101.22 VO Filterset
	301029	110,00	Karton 1170 x 770 x 750 mm	3-wellig Q.2.92
	302003	5 000,00	Zuschnitt	1500 x 40 mm, Q. 1.10
	302010	2 100,00	Wellpapp-Zuschnitt plano	780 x 1180 mm, Q. 1.30
	302013	400,00	Wellpapp-Zuschnitt	925 x 1210 mm, Q. 1.1
	302020	3 080,00	Wellpapp-Zuschnitt	1100 x 900 mm, Q. 1.0
	302025	800,00	Wellpapp-Zuschnitt plano	1193 x 250 mm Q. 2.7
	302026	1 500,00	Quersteg mittig geschlitzt	790 x 400 mm Q. 2.7
	302027	546,00	Wellpapp-6er-Gefache	1193x790x350 mm Q. 1.01
	302035	0,00	Schlitzverpackung flach	365x365x61 mm, Q. 1.02
	302039	1 290,00	Wellp. Gefach m. Blindfach	1180x390 + Blindfach 80 mm
	302040	550,00	Wellpapp-Ring Durchm.390	
	302041	0,00	J-Profil 1180 x 780 mm	Q. 1.02
	302042	420,00	Wellpapp-Zuschnitt Q. 1.01	1170x7070x760 (Gefache)
	302046	860,00	Abtrennung 1630 x 250 mm	passend zu 302041

Abbildung 6.8: Artikelkatalog mit integrierten ERP- und Bilddaten

Das steuernde Skript, das im Anhang 12.4 (Abbildung 12.21) dargestellt ist, beinhaltet den ODBC-Connect zur ERP-Datenbank (siehe VMI) sowie den Datei-Connect zur Dateistruktur der Bilddaten. Dadurch werden die erforderlichen Bildinformationen in die Webseite eingebunden. Aus beiden Systemen werden auf diese Weise die Informationen zusammengeführt und im Frontend in einer integrierten Oberfläche zur Verfügung gestellt. Damit ist der Artikelkatalog bereits in den wesentlichen Anforderungen abgebildet. Über ein zugriffssteuerndes Berechtigungskonzept wie im Falle des VMI ist nun noch innerhalb einer MySQL-Datenbank die Zuordnung von Teilen zu Kunden vorzunehmen, damit nur die für einzelne Kunden vorgesehenen Teile im Katalog enthalten sind. Hier sei auf die Ausführungen zum VMI verwiesen.

Im Rahmen der Portallösung sind noch weitere Integrationsszenarien bzgl. Daten und Dateien denkbar. Auf der Grundlage der bisherigen Szenarien wird leicht ersichtlich, dass die Integration von zahlreichen weiteren Prozessen stets auf die gleiche Art und Weise erfolgt. Das verwendete Prinzip ist stets die Einbindung externer Datenbanken und externer Dateistrukturen über Datenbank- und Datenkonnectoren, die über PHP-Funktionen integriert werden.

6.3.4 Prozessdatenaustausch / EDI

Während in den bisher beschriebenen Prozessen die Bereitstellung von Informationen für die Prozesspartner im Vordergrund stand, ist es im Sinne eines Real-Time Enterprises mitunter erforderlich, Prozessinformationen direkt an unternehmensexterne Systeme zur Weiterverarbeitung zu übertragen. Die darunter fallenden Prozesse werden im Rahmen des EDI zusammengefasst.

Wie bereits im Konzept in Kapitel 5 und bei der Realisierung des VMI beschrieben, ist die Generierung von XML-Nachrichten mittels LAMP eine wesentliche Voraussetzung, um eine direkte Prozesskommunikation zu ermöglichen. Ausgehend von dem bisher beschriebenen Integrationsszenario des VMI wird daher nachfolgend ein Prozess abgebildet, bei dem die VMI-Daten aus dem ERP-System als XML-Nachricht über eine Integrationsoberfläche einem anderen System zur Verfügung gestellt werden, um sie schließlich im entfernten System dispositiv in ein Warenwirtschaftssystem einzubinden.

Mittels PHP stehen für die Übertragung der XML-Nachrichten verschiedene Transportwege zur Verfügung. Die Übertragung kann beispielsweise als ASCII-Datei via ftp, http oder Email erfolgen oder auch in Form einer dedizierten Verbindung via smb-Protokoll. Voraussetzung für die Einbindung der XML-Daten im entfernten System ist der Abgleich der XML-Schemata auf beiden Seiten, d. h. die Definition, wie und in welchem Umfang die XML-Informationen verarbeitet oder eingebunden werden sollen. Diese Definition ist bei jeder Integrationslösung notwendig, die beteiligten Systeme stimmen dabei Art und Aufbau der übertragenen Informationen ab. Die Definitionen werden direkt in einem PHP-Skript umgesetzt, das gleichzeitig den Datenaustausch zwischen den Systemen steuert. An den System-Schnittstellen werden dabei in beiden Fällen LAMP-Umgebungen eingesetzt, die die XML-Nachrichten auf der internen Seite erzeugen und auf der externen Seite in das Zielsystem übertragen, das entweder selbst XML-fähig ist oder mittels LAMP in die Lage gebracht wird, XML-Daten zu verarbeiten.

Im Rahmen der Umsetzung werden im folgenden die ERP-Daten als XML-File aufbereitet und per Email versendet. In der Abbildung 6.9 ist die dafür verwendete Webseite dargestellt.

TTB Lieferantenportal

[XML-Export](#) [Logout](#)

Werk: 100

Bild	TTF-ArtNr	Bestand	Bezeichnung 1	Bezeichnung 2
	000169	360,00	Karton V 608900	545 x 465 x 450 mm IMC 110
	000435	2 000,00	Begleitpapiertaschen LDPE	240 x 160 + 25 mm, C5
	000778	0,00	Polybeutel	400 mm x 600 mm, 50 my
	000953	0,00	Punkt rot	Durchmesser 50 mm
	207004	4 600,00	Umreifungsband PT13 PB	13 mm, 0,6 mm, 2300 mtr.
	207009	3 000,00	Klammer für Umreifungsband	
	301005	200,00	Karton 370 x 370 x 350 mm	für 300.51/1/2/3/4/5
	301014	0,00	Faltschachtel A3 LL Votex	101.11 VO Filterset
	301015	0,00	Faltschachtel B3 LL Votex	101.22 VO Filterset
	301029	110,00	Karton 1170 x 770 x 750 mm	3-wellig Q.2.92
	302003	5 000,00	Zuschnitt	1500 x 40 mm, Q. 1.10
	302010	2 100,00	Wellpapp-Zuschnitt plano	780 x 1180 mm, Q. 1.30
	302013	400,00	Wellpapp-Zuschnitt	925 x 1210 mm, Q. 1.1
	302020	3 080,00	Wellpapp-Zuschnitt	1100 x 900 mm, Q. 1.0
	302025	800,00	Wellpapp-Zuschnitt plano	1193 x 250 mm Q. 2.7
	302026	750,00	Quersteg mittig geschlitzt	790 x 400 mm Q. 2.7
	302027	546,00	Wellpapp-Ger-Gefache	1193x790x350 mm Q 1.01
	302035	0,00	Schlitzverpackung flach	365x365x61 mm, Q. 1.02
	302039	1 290,00	Wellp. Gefach m. Blindfach	1180x390 + Blindfach 80 mm
	302040	550,00	Wellpapp-Ring Durchm.390	
	302041	0,00	J-Profil 1180 x 780 mm	Q. 1.02
	302042	420,00	Wellpapp-Zuschnitt Q. 1.01	1170x70/70x760 (Gefache)
	302046	860,00	Abtrennung 1630 x 250 mm	passend zu 302041

Abbildung 6.9: VMI mit XML-Datenexport

Darin sind die aus der VMI- bzw. Portalrealisierung erzeugten Webseiteninhalte aufgelistet (Artikelnummer, Lagerbestand, Bezeichnung und Bilddateien), die über einen XML-Export als XML-Nachricht per Email versendet werden können. Der Vorgang der Nachrichtenübertragung ist über eine Eventsteuerung gekoppelt, d. h., es ist eine Initiierung des Nachrichtenversandes per Mausklick erforderlich, der aber beispielsweise auch über einen Scheduling-Mechanismus zeitgesteuert getriggert sein kann.

Das dieser Lösung zugrundeliegende PHP-Skript enthält neben den Elementen der bereits beschriebenen Bereiche VMI und Portal zusätzlich noch den XML-Teil (vgl. Anhang 12.5 Abbildung 12.22). Die Basis für die XML-Daten sind die ERP-Daten, die über einen ODBC-Connect aus der DB2-Datenbank extrahiert werden (siehe VMI). Der Versand der Email wird mit einem PHP-Skript abgebildet (vgl. Anhang 12.6, Abbildung 12.23), indem die PHP-Funktion mail verwendet wird.

Damit stehen die XML-Daten am externen Zielsystem zur Verfügung, die dort entweder direkt eingelesen oder über LAMP in eine Datenbankstruktur zurückgeführt werden. Um im letztgenannten Fall die XML-Daten in das externe Zielsystem zu integrieren, erfolgt unter Verwendung der PHP-Funktion expat das Parsen der XML-Nachricht. Die so extrahierten

XML-Daten werden im Anschluss über den bekannten Pfad des nativen bzw. ODBC-Datenbankzugriffs in das Zielsystem eingebunden. Damit ist der komplette ERP-Integrationsprozess beschrieben, bei dem aus dem internen ERP-System die Datenintegration über XML an das externe System unter Verwendung der LAMP-Umgebung erfolgt.

6.4 Diskussion der erzielten Ergebnisse

Zur Umsetzung der Supply Chain Management-Szenarien VMI und ATP sowie der Portallösung und des Prozessdatenaustausches wurde in den vorangegangenen Kapiteln die LAMP-Umgebung als Integrationslösung hinsichtlich der an einer Supply Chain beteiligten IT-Systeme verwendet. Die Funktion der LAMP-Integrationslösung lag dabei in der Integration der relevanten Prozessdaten in die Systemumgebung der beteiligten Prozesspartner, die für das Supply Chain Management benötigt werden. Die Ergebnisse dieser Umsetzung werden im folgenden diskutiert, und zwar hinsichtlich der Erreichung des Integrationsziels sowie der Wirtschaftlichkeit der Integrationsanwendung im Hinblick auf den Vergleich von Kosten und Nutzen.

Für das VMI wurde von der mittelstandstypischen Situation einer heterogenen Systemlandschaft ausgegangen, in der die Lagerbestandsinformationen eines vorhandenen, aber nicht webbasiert integrierten ERP-Systems einem externen Lieferanten zur Verfügung gestellt werden sollten. Die beschriebene Umsetzung des VMI ermöglicht in einfacher Form die Darstellung von Lagerbestandsinformationen über einen Webbrowser, indem über eine ODBC-Datenbankschnittstelle direkt auf die Lagerbestandsinformationen zugegriffen wird. Hierfür wurden über ein LAMP-Integrationssystem die Lagerbestandsdaten des ERP-Systems extrahiert und anschließend im Webbrowser ausgegeben. Zentrale Integrationselemente bei dieser Umsetzung waren das steuernde PHP-Skript sowie der Apache Webserver, über den die Ausgabe der Bestandsdaten erfolgte.

Damit ist die wesentliche Funktion eines VMI, nämlich die Bereitstellung der Informationen über die Lagerbestände als Element der Verwaltbarkeit des Lagers durch den Lieferanten, erfüllt. Noch unberücksichtigt geblieben sind hier Prozesselemente, die es ermöglichen, zusätzliche Informationen zum VMI bereitzustellen, beispielsweise Informationen zu erfolgten Lieferungen oder erfolgten Lagerentnahmen. Im Rahmen der Anforderungen, die an ein VMI projektbezogen existieren, sind hierfür weitere ERP-Daten mittels PHP nach dem gleichen Schema, das hier beschrieben wurde, in die Browseransicht des Lieferanten einzubinden,

wodurch erweiterte Funktionalitäten für ein VMI im Rahmen einer Programmiererweiterung mittels PHP abgebildet werden.

Sofern eine funktionsfähige LAMP-Umgebung im betrachteten Unternehmen zur Verfügung steht, fallen im Rahmen der vorgestellten Lösung keine Systemkosten an und es ist im Rahmen der Umsetzung ausschließlich Programmieraufwand erforderlich. Die betrachtete programmtechnische Umsetzung liefert die elementaren Basisinformationen über die Lagerbestände im Rahmen eines VMI, die in zahlreichen Anwendungsfällen ausreichend sind. Grundsätzlich ist der darüber hinaus entstehende Aufwand von den funktionalen Projektanforderungen an ein VMI sowie von dem zu integrierenden ERP-System abhängig. Durch die umfangreiche Unterstützung einer Vielzahl von Datenbankstrukturen mittels PHP ist in einer Vielzahl von Integrationsfällen die Einbindung der Datenstrukturen des betreffenden ERP-Systems möglich.

Die aus Gründen des Investitionsschutzes zu betrachtende Situation der Administrierbarkeit des vorgestellten VMIs stellt sich sehr positiv dar, denn die Erweiterung des VMI zu einer Mehrmandanten-Struktur ist bereits im Rahmen der programmtechnischen Umsetzung via datenbankgesteuertem Berechtigungskonzept gegeben. Zur Erweiterung der Berechtigungsstruktur ist insofern lediglich die Bestückung der verwendeten MySQL-Datenbanktabellen mit Datensätzen erforderlich, die Anmeldeinformationen und Lagerzuordnungen für die anzubindenden Lieferanten enthalten. Für diese Erweiterung steht unter Linux beispielsweise das Tool PHPMYAdmin zur Verfügung, mit dem MySQL-Datenbanken und Datenbanktabellen über ein Webformular verwaltet werden können. Dadurch ist der erforderliche Aufwand bei einer Ausweitung des VMI auf zusätzliche Lieferanten nur noch gering.

Infolge der geringen Kostenstruktur beim betrachteten VMI ist die besondere Eignung für die mittelständische Automotive-Industrie gegeben. Die LAMP-Umgebung stellt sich hier als pragmatisches Integrationskonzept dar, das vollständig skalierbar und im Rahmen der jeweiligen Anforderungen erweiterbar ist. Bei Betrachtung der Nutzenpotentiale der VMI-Lösung aus Kapitel 5 lassen sich berechenbare bzw. abschätzbare Verbesserungen der Kostenstruktur ableiten. Unter der Annahme, dass in der Bilanzstruktur des verarbeitenden Gewerbes das Umlaufvermögen mit etwa 50% auf der Aktiva-Seite bewertet wird [127], ergibt sich bei einer Reduzierung des Umlaufvermögens um 20% gemäß Andersen-Consulting und einem Umsatz von 60 Mio. EUR/a eine Senkung des Umlaufvermögens um 6 Mio. EUR. Bei einer Eigenkapitalquote im Mittelstand von 30% verbleibt damit ein Finanzierungsbedarf von 4 Mio. EUR, der zinsbewertet mit 5%/a einen Kapitalkaufwand von 200.000,00 EUR/a darstellt. Im Vergleich zu den Kosten des VMI, die bei der Umsetzung

mittels LAMP entstehen, ist selbst bei pessimistischer Schätzung eine Amortisation bereits nach kurzer Zeit erreicht.

Die programmtechnische Umsetzung des ATP-Szenarios basiert im Grunde genommen auf den Grundlagen des VMI, das zur Abbildung der ATP-Spezifika funktional erweitert wurde. Der wesentliche Unterschied ist, dass zusätzlich zur ERP-Datenbank weitere externe Daten aus einem anderen System integriert wurden (Feinplanungssystem mit Oracle-Datenbank). Das Ergebnis der Umsetzung ermöglicht es dem Kunden, in einer Kunden-Lieferanten-Beziehung den verfügbaren Lagerbestand sowie die laufende und künftig geplante Teilefertigung seines Lieferanten einzusehen. Dadurch wird der Kunde in die Lage versetzt, die Erfüllbarkeit seiner Bedarfe abzuschätzen und seine eigene Disposition zielgenauer zu steuern.

Wie im Falle des VMI sind auch bei der Verfügbarkeits- bzw. Lieferbarkeitsprüfung in der ATP-Umsetzung nur die Grundelemente abgebildet, so dass projektbezogen auch hier die PHP-Programmierung entsprechend den Anforderungen zu erweitern ist. Der im Rahmen der Umsetzung erforderliche Programmieraufwand auf der Basis der LAMP-Umgebung erstreckt sich im vorliegenden Fall auf die funktionale Ausweitung des für das VMI beschriebenen PHP-Skriptes. Der aus dieser Umsetzung resultierende Nutzen stellt sich auf Kundenseite hauptsächlich durch eine genauere Planung der Supply Chain dar. Für den Lieferanten ergibt sich neben einer verbesserten Liefertreue ein Nutzen dadurch, dass infolge frühzeitig eingehender Bestellungen die Produktion zeitnaher erfolgt und die produzierte Ware schneller zur Auslieferung gebracht wird, was einer Verkürzung der Liefer- und Lagerzeiten auf der Seite der Rohstoffe und der Fertigware gleichkommt. Parallel dazu erhöht sich der Servicegrad des Lieferanten gegenüber seinen Kunden. Eine exakte Nutzenschätzung dieser Faktoren ist infolge der Bewertungsproblematik (Unschärfe bzgl. der Bewertungskriterien) nur schwer durchführbar.

Bei der Portallösung, die für viele SCM-Prozesse das Frontend auf Kunden- oder Lieferantenseite bildet, ist der Integrationsaufwand am größten. In der Portallösung werden verschiedenste Informationen über Lagerbestände, Artikel oder Lieferungen zur Verfügung gestellt, wobei die hierfür erforderlichen Daten aus vielen unterschiedlichen Systemen stammen. Um die Heterogenität einer solchen Struktur in der Portallösung zu beseitigen, muss für jedes dieser Systeme ein Integrationspfad mittels LAMP zur Verfügung gestellt werden. Im Rahmen der Realisierung wurden neben den Datenbankintegrationen aus dem VMI- und dem ATP-Prozess auch Dateiintegrationen mit externen Dateistrukturen durchgeführt. Da das beschriebene Integrationsprinzip keine Einschränkungen bzgl. der Anzahl der zu integrierenden Systeme besitzt, ist es auch für umfangreiche Portallösungen

anwendbar und infolge der weitreichenden Datenbankunterstützung von PHP auch gut geeignet. Die Programmierung mittels PHP kann dabei problemlos in vorhandene Portalstrukturen eingebunden werden. Demgegenüber stellt der übliche nach Analystenschätzungen erforderliche Aufwand für den Portalaufbau mit Kosten ab 50.000,- EUR eine beachtliche Größenordnung dar [128].

Mit der Abbildung von EDI-Prozessen bzw. Prozessdatenkommunikation wird eine eigene Kategorie bei der Betrachtung von SCM-Szenarien beschrieben. Hier wechselt die Perspektive von der reinen Informationsdarstellung zur interprozessualen Anwendungskommunikation. Die vorgestellte Prozessdatenkommunikation mittels XML-Nachrichten auf der Basis der LAMP-Umgebung ermöglicht es dabei, einen standardisierten Datenaustausch aus Anwendungen, die selbst nicht XML-fähig sind, umzusetzen, indem zur Einbindung der benötigten Daten sowie zur Prozesssteuerung PHP-Funktionen innerhalb von PHP-Skripten verwendet werden. Grundsätzlich ist damit die Unterstützung jedes Geschäftsprozesses möglich, in dem die Kommunikation auf der Basis von XML stattfindet, allerdings steigt der Programmieraufwand mit der Komplexität der Prozesse schnell an. Im realisierten Geschäftsprozess wurde die Übertragung von Lagerbestandsinformationen abgebildet, die es ermöglicht, einem externen System diese Informationen in Form einer XML-Nachricht zur Verfügung zu stellen.

Der resultierende Nutzen ist maßgeblich vom betrachteten Geschäftsprozess abhängig. Im vorliegenden Fall ist in der Prozessautomatisierung der größte Wert zu sehen, da sich der von außen manuell einzutragende Prozessanteil durch die Prozessautomatismen reduziert und darüber hinaus die Qualität der verarbeitenden Daten ansteigt, was wiederum zu einer verringerten Fehlerquote im Gesamtprozess führt. Unterzieht man nur diejenigen Faktoren einer Bewertung, die aus einer Reduzierung der Prozesszeit resultieren, so ergibt sich die nachstehende Berechnung: bei einem angenommenen manuellen Bearbeitungsaufwand für nicht automatisiert verarbeitete Bestellungen von 1 h/Tag beträgt die Reduzierung der Prozesszeit in erster Näherung 240 h/a. Bei durchschnittlichen Arbeitskosten von 25,- EUR/h als Mischkalkulation von produzierendem Gewerbe und ausgewählten Dienstleistungsbereichen [129] ergibt sich hieraus eine jährliche Prozesskostenreduzierung von 6.000,- EUR. Allein aus dieser Perspektive heraus liegt der Amortisationszeitraum für den betrachteten Prozessdatenaustausch im Bereich weniger Monate. Für komplexe Geschäftsprozesse kann jedoch der Programmieraufwand die Amortisationsberechnung umkehren und die Wirtschaftlichkeit in Frage stellen. Insofern ist für jeden betrachteten Geschäftsprozess eine eigene Wirtschaftlichkeitsanalyse bzw. Kostenschätzung durchzuführen [130].

Insgesamt lässt sich im Ergebnis feststellen, dass sich unter Verwendung der LAMP-Umgebung mit PHP als funktionalem Integrationskernelement die betrachteten SCM-Szenarien und -Prozesse in einfacher Form umsetzen lassen, für die sonst teure EAI-Systeme auf der Grundlage von Application Servern eingesetzt werden müssten. Hier bietet die LAMP-Struktur durch den Einsatz von PHP leistungsfähige und kostengünstige Lösungsansätze, die für den automobilen Mittelstand gut geeignet sind, wenngleich im Hinblick auf die Prozesskomplexität fallspezifische Einschränkungen zu berücksichtigen sind.

7 Zusammenfassung und Ausblick

Veränderte Märkte und sich verschärfende Wettbewerbsbedingungen stellen neue Herausforderungen an die Automobilindustrie, die mit effizienteren Prozessstrukturen den Veränderungen begegnen muss. Erfolgversprechende Konzepte werden dabei im Rahmen des Supply Chain Managements zur Verfügung gestellt, worunter die integrierte prozessorientierte Planung und Steuerung der Waren-, Informations- und Geldflüsse entlang der gesamten Wertschöpfungskette vom Rohstofflieferanten bis zum Kunden verstanden wird, mit den Zielsetzungen einer verbesserten Kundenorientierung, einer Synchronisation der Versorgung mit dem Bedarf, einer flexiblen und bedarfsgerechten Produktion sowie einem Abbau der Bestände entlang der Wertschöpfungskette.

Die bisher zur Steuerung der Supply Chain eingesetzten IT-Systeme innerhalb eines Unternehmens sind zumeist Bestandteil einer heterogenen Systemwelt und durch Inkompatibilitäten daran gehindert, untereinander die erforderlichen Daten auszutauschen. Diese Heterogenität stellt eine Hürde bei der Umsetzung von SCM-Konzepten dar, deren Implementierung wiederum erforderlich ist, um die in der Automobilindustrie durch Marktveränderungen erzwungene Effizienzsteigerung in der Supply Chain zu ermöglichen. Auf dem Weg zu einer unternehmensübergreifenden Kommunikation ist daher die Integration der im Unternehmen eingesetzten Anwendungssysteme eine notwendige Voraussetzung, um die erforderlichen Prozessdaten den beteiligten Prozesssystemen zur Verfügung zu stellen. Diese Integrationsaufgaben werden von EAI-Systemen übernommen.

Im Rahmen dieser Arbeit wurden die marktführenden EAI-Systeme betrachtet und ihre grundlegenden Eigenschaften dargestellt. Neben einem umfangreichen Leistungsspektrum ist allen Systemen gemein, dass sie teuer und aufwendig zu implementieren sind, wodurch sie infolge eines hohen Lizenz-, Projektierungs- und Administrationsaufwandes nicht den mittelstandstypischen Kostenstrukturen entsprechen. Daher sind andere EAI-Lösungen erforderlich, mit denen auf einfache und kostengünstige Art und Weise die Umsetzung von SCM-Konzepten ermöglicht werden kann. Hierzu eignet sich die für Webtechnologien bereits häufig verwendete LAMP-Systemstruktur, die ein Gesamtsystem aus den als kostenfreie open source-Software zur Verfügung stehenden Komponenten Linux, Apache, MySQL und PHP bildet. Die LAMP-Umgebung, die in vielen Fällen bereits für die Bereitstellung von Webservices in mittelständischen Unternehmen vorhanden ist, besitzt durch die strenge Ausrichtung an Webtechnologien schon die Voraussetzungen für eine webbasierte Anwendungsintegration und stellt die Verbindungen zwischen einzelnen Unternehmensanwendungen bzw. -prozessen her. Zentrales Integrationselement ist dabei

die Skriptsprache PHP, über die die benötigten Integrationsfunktionalitäten abgebildet werden.

Im Rahmen dieser Arbeit wurden mit LAMP vier SCM-Szenarien beschrieben und schematisch umgesetzt, wobei ein VMI, ein ATP-Szenario, eine Portallösung sowie ein anwendungsübergreifender Prozessdatenaustausch auf der Basis einer standardisierten XML-Kommunikation ausgearbeitet wurden. In allen beschriebenen Szenarien erfolgte die Anwendungsintegration über die Integration der Daten, die entweder aus Datenbank- oder Dateistrukturen unterschiedlicher Anwendungssysteme stammen. Je nach SCM-Szenario variiert in der betrachteten Systemumgebung die Anzahl der zu integrierenden Systeme, wodurch im Wesentlichen der Aufwand bestimmt wird, der zum Integrationsziel führt. Im Ergebnis bleibt festzuhalten, dass für die vorgestellten SCM-Szenarien im Rahmen einfacher Anforderungen mit LAMP eine leistungsfähige und kostengünstige Plattform zur Verfügung steht, um die Basisfunktionalitäten in den abgebildeten SCM-Prozessen bereitzustellen. Für komplexe Prozesse ist der entstehende Programmieraufwand gegenüber dem Nutzen im Einzelfall stets neu zu betrachten, damit die Wirtschaftlichkeit der Programmierleistung gegenüber dem erwarteten Nutzen gewahrt bleibt.

Künftige Entwicklungsrichtungen bei der Verwendung von LAMP als Integrationsplattform liegen in der Vereinfachung der Möglichkeiten zur Prozessbeschreibung und damit im Bereich neutraler Business-Objekte, über die beispielsweise eine Wiederverwendbarkeit von Prozesselementen oder ein einfaches Datenmapping hinsichtlich der Verwendung von Repositories oder XML-Schemata ermöglicht wird. Beispielsweise wäre dies durch eine verbesserte Objektorientierung der Fall. Nicht LAMP-spezifisch, sondern als generelles Problem webbasierter Integrationsmodelle, stellt sich das Fehlen von Security-Standards dar. Es gibt zwar vielversprechende Vorlagen aus Industriekonsortien in wechselnder Besetzung: WS-Secure Conversation, WSPolicy, WS-Trust sowie WS-Federation, WS-Authorization und WS-Privacy, aber die Freigabe der maßgeblichen Standardisierungsgremien wie des W3C oder der **OASIS**¹⁵² haben diese Vorschläge noch nicht.

Eine an Bedeutung gewinnende Funktionalität von PHP, die im Rahmen der Anwendungsintegration eine wesentliche Rolle spielt, ist in der Anbindung an Komponentenmodelle zu sehen, die auf einer standardisierten Programmstruktur beruhen, z. B. CORBA. CORBA ist eine Antwort auf die starke Zunahme von Hardware- und Software-Produkten und Ziel war es, eine Middleware zu schaffen, welche eine orts-, plattform- und

¹⁵² Organization for Advancement of Structured Information Standards

implementationsunabhängige Kommunikation zwischen Applikationen erlaubt [131]. Mit CORBA wird eine nahtlose Interoperabilität zwischen Applikationen erreicht, welche in einem völlig heterogenen Umfeld betrieben werden können. Neben CORBA existieren noch zwei weitere Architekturen für verteilte Objekte, nämlich das DCOM von Microsoft und das DCE [132]. Die Unterstützung dieser Komponentenmodelle steht bereits für PHP in Form von Erweiterungen zur Verfügung, so dass darüber der Zugriff auf entfernte Objekte möglich ist. Damit besitzen PHP und die LAMP-Systemstruktur das Potential, die Funktionalität vollintegrierender EAI-Systeme abzudecken und sich zukunftsweisend in heterogene Systemlandschaften einzufügen.

8 Abbildungsverzeichnis

Abbildung 1.1: IT-Szenario in der Automobilindustrie ohne EAI	2
Abbildung 1.2: EAI-Plattform zur Integration der IT-Anwendungen	3
Abbildung 2.1: Die Automobilindustrie im Wandel [2]	6
Abbildung 2.2: Prozessoptimierung versus Funktionsoptimierung [8].....	8
Abbildung 2.3: Veränderungen der Erzeugerpreise in der Automobilindustrie [18]	12
Abbildung 2.4: Prozesspfad bei Hersteller-Lieferanten-Zusammenarbeiten [30].....	15
Abbildung 2.5: Veränderung von Anzahl und Umsatz der Zulieferer [9]	19
Abbildung 2.6: Entwicklungsstufen zwischen OEM und Zulieferer [63]	22
Abbildung 2.7: Gewinneffekte bei Materialkostenreduzierung und	25
Abbildung 2.8: Bull-Whip-Effekt [83].....	26
Abbildung 2.9: Variantenvielfalt der eingesetzten CA-Systeme am Beispiel BMW [30].....	29
Abbildung 2.10: Anwendungssysteme in der logistischen Kette	30
Abbildung 2.11: IT-Systemlandschaft mit nur teilweise verbundenen Anwendungen.....	34
Abbildung 2.12: heutige Situation im mittelständischen Automotive-Bereich.....	37
Abbildung 3.1: externe Integration zwischen Unternehmen [99].....	40
Abbildung 3.2: interne Integration zwischen Anwendungen [99].....	40
Abbildung 3.3: interne, externe und integrierte Supply Chain [3]	40
Abbildung 3.4: Integrationsgrad von EAI und traditioneller Integrationsansätze [99].....	41
Abbildung 3.5: Integrationsszenario für interne und externe Integration [100].....	42
Abbildung 3.6: EAI-Referenzstruktur [101].....	45
Abbildung 3.7: Integrationsmethoden für die Anwendungsintegration [101]	46
Abbildung 4.1: Marktanteile EAI-Anbieter, USA 2001 [99]	50
Abbildung 5.1: ActiveEnterprise Architektur [100]	57
Abbildung 5.2: AllianceManager Architektur [100]	60
Abbildung 5.3: Architektur des Microsoft BizTalk Server 2002 [98]	62
Abbildung 5.4: BusinessWare Architektur [100]	65
Abbildung 5.5: Architektur von Constellar Hub [100].....	68
Abbildung 5.6: Interchange Server als Bestandteil der CrossWorlds Architektur [100]	70
Abbildung 5.7: Mapping von realen Businessobjekten auf.....	70
Abbildung 5.8: Transaktionen bei CrossWorlds für langlebige Prozesse [100].....	71
Abbildung 5.9: Architektur von e*Gate / e'Xchange [100]	73
Abbildung 5.10: e-Biz Integrator [100].....	75
Abbildung 5.11: eLink Plattform [100]	78
Abbildung 5.12: Forté Fusion Architektur [100]	80
Abbildung 5.13: iWave Architektur [100]	82

Abbildung 5.14: MINT Knowledge Family Architektur [100].....	86
Abbildung 5.15: Hub and Spoke Architektur mit MQIntegrator [100]	88
Abbildung 5.16: Architektur CandleNet eBusiness Plattform, vormals ROMA [100].....	90
Abbildung 5.17: WebMethods Enterprise Architektur [100].....	93
Abbildung 5.18: Total Cost of Ownership bei Linux, Solaris und Windows in US \$ [109]	101
Abbildung 5.19: PHP-Syntax am Beispiel des Skriptes hello.php.....	102
Abbildung 5.20: MySQL-Datenbankanbindung mit PHP	103
Abbildung 5.21: LAMP-Integrationsschema	105
Abbildung 5.22: EAI-Systemstruktur mit LAMP	107
Abbildung 5.23: Teilbereiche des Supply Chain Managements [3].....	108
Abbildung 5.24: Situation der Beziehungen im EAI-LAMP-Szenario	110
Abbildung 5.25: Verkürzung der Auftragsabwicklung mit und ohne VMI [116]	113
Abbildung 5.26: Prozessstruktur beim VMI	114
Abbildung 5.27: LAMP-Integrationselemente beim Vendor Managed Inventory.....	116
Abbildung 5.28: Prozessstruktur bei ATP.....	118
Abbildung 5.29: LAMP-Integrationselemente bei ATP	119
Abbildung 5.30: Prozessstruktur bei einer Portallösung.....	121
Abbildung 5.31: LAMP-Integrationselemente bei Portallösungen	122
Abbildung 5.32: Kommunikation zwischen ERP-Anwendungen via XML	124
Abbildung 5.33: LAMP-Integrationselemente bei der XML-Kommunikation	125
Abbildung 6.1: Software- und Systemumgebung des betrachteten Unternehmens.....	129
Abbildung 6.2: Konfigurationsdatei odbc.ini	132
Abbildung 6.3: Loginseite eines Lieferanten für das VMI	132
Abbildung 6.4: Datenbanktabellen der Datenbanken MySQL und DB2 für das VMI	134
Abbildung 6.5: Lagerbestandsinformationen aus dem ERP-System im Browserfenster des Lieferanten	135
Abbildung 6.6: Ablaufdiagramm zum VMI	136
Abbildung 6.7: Ablaufdiagramm für die artikelspezifischen Inhalte	138
Abbildung 6.8: Artikelkatalog mit integrierten ERP- und Bilddaten	140
Abbildung 6.9: VMI mit XML-Datenexport	142
Abbildung 12.1: Tabellenbeschreibung der Datenbanktabelle auth.....	171
Abbildung 12.2: Tabellenbeschreibung der Datenbanktabelle rights	171
Abbildung 12.3: Konfigurationsdatei conf.php.....	172
Abbildung 12.4: Konfigurationsdatei conf.odbc.php	172
Abbildung 12.5: Konfigurationsdatei conf.mysql.php	172
Abbildung 12.6: Konfigurationsdatei head.php.....	173
Abbildung 12.7: Konfigurationsdatei foot.php.....	173
Abbildung 12.8: Beginn der Programmdatei index.php (Teil 1).....	174

Abbildung 12.9: Programmdatei index.php (Teil 2)	174
Abbildung 12.10: Programmdatei index.php (Teil 3)	175
Abbildung 12.11: Programmdatei index.php (Teil 4)	175
Abbildung 12.12: Programmdatei index.php (Teil 5)	176
Abbildung 12.13: Programmdatei index.php (Teil 6)	177
Abbildung 12.14: Programmdatei index.php (Teil 7)	178
Abbildung 12.15: Programmdatei index.php (Teil 8)	178
Abbildung 12.16: Programmdatei index.php (Teil 9)	179
Abbildung 12.17: Programmdatei index.php (Teil 10)	179
Abbildung 12.18: Programmdatei oci.php (Teil 1)	180
Abbildung 12.19: Programmdatei oci.php (Teil 2)	180
Abbildung 12.20: Programmdatei oci.php (Teil 3)	181
Abbildung 12.21: Programmskript zur Einbindung von Bilddateien	182
Abbildung 12.22: PHP-Skript zur Ausgabe der Datenbankinhalte als XML-Nachricht	183
Abbildung 12.23: PHP-Programmskript zum Versand	184

9 Tabellenverzeichnis

Tabelle 1: Zusammenfassung des Ist-Zustandes	35
Tabelle 2: Architekturen der EAI-Systeme	53
Tabelle 3: Bewertung der EAI-Systeme	97
Tabelle 4: LAMP-Umgebung in dem Pilot-Unternehmen	130
Tabelle 5: Installierte Software-Systeme in dem Pilot-Unternehmen	130
Tabelle 6: Tabelle auth der Datenbank konsi.....	133
Tabelle 7: Tabelle rights der Datenbank konsi	133

10 Literaturverzeichnis

- [1] Serain, D.: Middleware and enterprise application integration: the architecture of e-business solutions, 2nd ed., Springer Verlag, London 2002
- [2] Hägele, T., Schön, W. U.: Erfolgsstrategien für Automobilzulieferer, in: ZWF, 93 (1998) 7-8
- [3] Lawrenz, O., Hildebrand, K., Nenninger, M., Hillek, T.: Supply Chain Management, 2. Auflage, Vieweg Verlag, Braunschweig/Wiesbaden, 2001
- [4] Gaul, H-D.: Verteilte Produktentwicklung – Perspektiven und Modell zur Optimierung, Dissertation TU München, Fakultät für Maschinenwesen, 2001
- [5] Hungenberg, H., Wulf, T.: Business Process Engineering, in: ZWF, 93 (1998) 7-8
- [6] Hammer, M.: Das prozessorientierte Unternehmen, München: Heyne 1999
- [7] Hammer, M., Champy, J.: Reengineering the Corporation. A Manifesto for Business Revolution. New York: Harper Business 1993
- [8] Thaler, K.: Supply Chain Management, 3. Auflage, Fortis Verlag, Köln, 2001
- [9] Bain & Company: Wachstum aus dem Kern – Strategische Herausforderungen für die Automobilzulieferindustrie, München, 2002
- [10] Womack, J. P., Jones, D. T., Roos, D.: Die zweite Revolution in der Automobilindustrie. Konsequenzen aus der weltweiten Studie des Massachusetts Institute of Technology, 8., durchgesehene Auflage, Frankfurt/New York 1994
- [11] Hebeisen, W.: F. W. Taylor und der Taylorismus, 1. Auflage, vdf Hochschulverlag, 1999
- [12] Lang, K.: Herausforderungen und Handlungsmöglichkeiten, 2. Auflage, Bund-Verlag, Köln 1994

-
- [13] Hans-Böckler-Stiftung: Lean production: Kern einer neuen Unternehmenskultur und einer innovativen und sozialen Arbeitsorganisation ?, 1. Auflage, Nomos-Verlagsgesellschaft, Baden-Baden 1992
- [14] Schögel, M., Sauer, A.: Auf dem Weg zur Multichannel Distribution – Herausforderungen für die Automobilbranche Handelsforschung 2001/02, Wandel der handelsrelevanten Wertschöpfungsprozesse, Trommsdorff, V., Köln
- [15] AFW Wirtschaftsakademie Bad Harzburg, Unternehmensführung I, 2004
- [16] Köpf, P.: Stichwort: Globalisierung, München, Heyne, 1998
- [17] Fechner, M.: Der Marktauftritt japanischer Automobilhersteller in Südostasien Dissertation Universität Göttingen, Fachbereich Wirtschaftswissenschaften, 2000
- [18] Statistisches Bundesamt, 2001
- [19] Planning Perspectives Inc.: North American Automotive Supplier Survey 2001, Birmingham 2001
- [20] Financial Times Deutschland v. 13.08.2001
- [21] GVO, Europäische Kommission 2000
- [22] Dudenhöffer, F.: Restrukturierung von Vertriebsnetzen C. Belz & T. Tomczak (Hrsg.), Thexis, Marktbearbeitung und Distribution, Kompetenz für Marketing-Innovationen, 1997, Schrift 4, St.Gallen: Thexis
- [23] Soliman, P., Otto, U.: Die Zukunft des Automobilvertriebs. Schriftenreihe der Unternehmensberatung Booz-Allen-Hamilton zu Vertriebsstrategien, Frankfurt, 2000
- [24] Statistisches Bundesamt, 2002
- [25] o. V.: 2005: 40 Prozent Überkapazität. Prognose von Ford-Präsident W.W. Boddie, Autohaus, Ausgabe 10/98
- [26] Corsten, D., Gabriel, C.: Supply Chain Management erfolgreich umsetzen: Grundlagen, Realisierung, Fallstudien, Springer Verlag, Berlin Heidelberg 2002

- [27] Eversheim, W. (HRSG., 1995B): Prozessorientierte Unternehmensorganisation, Berlin: Springer 1995.
- [28] Reinhart, G.; Lindemann, U.; Heinzl, J.: Qualitätsmanagement, Berlin: Springer 1996.
- [29] Osterloh, M.; Frost, J.: Prozessmanagement als Kernkompetenz, Wiesbaden 1996
- [30] Heftrich, F.: Moderne F&E-Zusammenarbeiten in der Automobilindustrie – Organisation und Instrumente – Dissertation Universität-Gesamthochschule Siegen, Fachbereich Wirtschaftswissenschaften, 2000
- [31] FAZ vom 10/02/03
- [32] Grabowski, H.; Geiger, K.: Neue Wege zur Produktentwicklung, Stuttgart, Raabe, 1997
- [33] Späth, L.: Zukunft ohne Risiko? Haben wir Angst vor zuviel Wandel? In: Reinhart, G.; Hoffmann, H. (Hrsg.): Nur der Wandel bleibt. Wege jenseits der Flexibilität. Münchener Kolloquium 2000, München: Utz 2000
- [34] Gausemeier, J.; Lindemann, U.; Reinhart, G.; Wiendahl, H.-P.: Kooperatives Produktengineering, Paderborn, Heinz Nixdorf Institut 2000, HNI-Verlagsschriftenreihe
- [35] Haak, R.: Kooperationsmanagement der japanischen Industrie in fortschrittlichen Technologiefeldern, In: Industrie Management, 16 (2000) 6
- [36] Gausemeier, J.; Möhringer, S.; Wleklinski, C.: Kooperatives Produktengineering. Ein neues Selbstverständnis des ingenieurmäßigen Wirkens, In: ZWF, 95 (2000) 10
- [37] Eversheim, W.; Laufenberg, L.; Heyn, M.; Kümpfer, R.; Linnhoff, M.; Schares, L.; Heyn, W.; Bohr, R.P.; Ehring, R.; Vorweg, M.: Entwicklung von Fahrzeugsystemen im Verbund, In: VDI-Z, 137 (1995) 5
- [38] Dudenhöffer, F.: Konzentrationsprozesse in der Automobilindustrie: Stellgrößen für die Rest-Player, In: Zeitschrift für Betriebswirtschaft, Heft 4/01
- [39] Wildemann, H.: Interne und externe Kunden-Lieferanten-Beziehungen, In: ZWF, 93 (1998) 1-2

- [40] Lange, C.: Der Automobilsektor in Südostasien: Strategieempfehlungen für deutsche Pkw-Hersteller: Eine Analyse unter Berücksichtigung der globalen Wettbewerbssituation, Kaiserslautern 1995
- [41] Nolte, D. : Freihandel, Protektion oder «Fair-Trade» auf den Weltautomobilmärkten? Analyse von struktureller Entwicklung und Wachstumspotentialen der Pkw-Industrie, insbesondere der Volumenhersteller in der Bundesrepublik Deutschland, Thun, Frankfurt am Main 1990
- [42] Dudenhöffer, F: Markenmanagement bei Produkt-Konvergenz. Neue Ansätze im Automobilmarketing, In: Jahrbuch der Absatz- und Verbrauchsforschung, Heft 1/1997 (Sonderdruck) Berlin, Duncker & Humblot
- [43] Frankfurter Allgemeine Zeitung vom 29.03.1999
- [44] Wildemann, H.: Entwicklungsstrategien für Zulieferunternehmen, 2. Aufl. München 1995
- [45] Demant, H. H.: Fair und effektiv, In: Automobil-Entwicklung, 1994
- [46] Hägele, T.; Schön, W.-U.: Erfolgsstrategien für Automobilzulieferer, In: ZWF, 93 (1998) 7-8
- [47] Wirth, S.: Logistikplanung für vernetzt kooperierende Fabriken, In: Fachtagung Planung von Logistiksystemen 95. Kooperation und Integration, Magdeburg: Inst. für Förder- und Baumaschinentechnik, Stahlbau, Logistik ,1995
- [48] Jansen, R.: Innovationen in der Logistik – Chancen in einem dynamischen Umfeld, In: Log-In 1999 – vom Kostendruck zur Innovationsdynamik, 9. Deutscher Materialflussskongress, Düsseldorf: VDI-Verlag, 1999
- [49] Müller-Stewens, G./ Gocke, A.: Kooperation und Konzentration in der Automobilindustrie, Amsterdam 1995
- [50] Meffert, H.: Trends im Konsumentenverhalten – Implikationen für Efficient Consumer Response, Ahlert, D.; Borchert, S. [2000], Prozessmanagement im vertikalen Marketing, Berlin, Heidelberg, New York, Springer

- [51] Adolphs, B.: Stabile und effiziente Geschäftsbeziehungen: Eine Betrachtung von vertikalen Koordinationsstrukturen in der deutschen Automobilindustrie, Lohmar, Köln 1996
- [52] Burckhardt, W.: Wandel im Unternehmen: Vernetztes und kundenorientiertes Gestalten von Produkten und Unternehmen im Umfeld nichtlinearer Veränderungen, Münchner Kolloquium '94, Landsberg/Lech, Verlag mi, 1994
- [53] Bänsch, A.: Variety Seeking – Marketingfolgerungen und Untersuchungen zum Abwechslungsbedürfnis von Konsumenten, In: Jahrbuch für Absatz- und Verbraucherforschung, Heft 4/95
- [54] PIEPER, W.: Der MOHAG Auto-Fachmarkt. Auto & Mehr - Gegenwart & Zukunft – Strategie & Chance, Ottobrunn, Autohaus, 1997
- [55] Diez, W.: Autos bei Tchibo? Die Zukunft des selektiven Vertriebs in der Automobilbranche, Arbeitspapier Nr. 2/2000 des IFA der Fachhochschule Nürtingen
- [56] GROSS, P.: Die Multioptionsgesellschaft, Frankfurt a. M., Suhrkamp
- [57] Schögel, M.: Mehrkanalsysteme in der Distribution, Wiesbaden, Gabler/DUV, 1997
- [58] Reinhart, G.; Kress, M.: Mass Customization Based on Constraint-Space Product and Process Modelling, In: Lindemann, U.; Birkhofer, H.; Meerkamm, H.; Vajna, S. (Hrsg.): Proceedings of the 12th International Conference on Engineering Design. Volume 1, München, TU, Eigenverlag 1999
- [59] Wildemann, H.: Effektives Variantenmanagement, In: ZWF, 94 (1999) 4
- [60] Meissner, M./Mehrle, N.: Automobilvertrieb im Internet. Beiträge zur Marketingwissenschaft, Nr. 31 (Herausgegeben vom Lehrstuhl Prof. Dr. G. Silberer, Institut für Marketing und Handel), Universität Göttingen, 2000
- [61] Schögel, M./Birkhofer, B./Tomczak, T.: E-Commerce im Distributionsmanagement, Fachbericht für Marketing 2000/2. St. Gallen: Thexis.
- [62] Accenture: Branchenstudie Auto 2010, Sulzbach/Taunus: Accenture GmbH, 2001.

- [63] Jörgensen, S.: Collaborative Engineering – Anforderungen und Trends in der Zusammenarbeit OEM – Zulieferer in der Automobilindustrie, IBM-Forum Stuttgart, Januar 2003
- [64] Schlüter, K.: Nutzungsgradsteigerung von Montagesystemen durch den Einsatz der Simulationstechnik, München, Hanser 1989
- [65] Reinhart, G.: Innovative Prozesse und Systeme – Der Weg zu Flexibilität und Wandlungsfähigkeit, Münchner Kolloquium `97, Landsberg/Lech: Verlag mi 1997
- [66] Lindemann, U.: Erfolgreiche Produkte durch Integrierte Produktentwicklung, Münchner Kolloquium `97, Landsberg/Lech, Verlag mi 1997
- [67] Pfohl, H.-Ch., Häusler, P., Koldau, A.: Qualität distributionslogistischer Leistungen, Jahrbuch der Logistik 1999, Düsseldorf, Verlagsgruppe Handelsblatt 1999
- [68] Wiendahl, H-P.: Fertigungsregelung: Logistische Beherrschung von Fertigungsabläufen auf Basis des Trichtermodells, München, Hanser 1997
- [69] Milberg, J.: Produktion – Eine treibende Kraft für unsere Volkswirtschaft, Münchner Kolloquium `97, Landsberg/Lech, Verlag mi, 1997
- [70] Jetter, M.: E-Business schließt Lücken der Fertigungsindustrie, VDI-Nachrichten, 24/09/99, Nr. 38, Düsseldorf, VDI-Verlag 1999
- [71] Günthner, W. A., Haller, M.: Im Spannungsfeld zwischen Flexibilität und Automatisierung, Jahrbuch der Logistik 1999, Düsseldorf, Verlagsgruppe Handelsblatt 1999
- [72] Günthner, W. A.: Vorlesungsskript Fördertechnik I, München, Lehrstuhl fml, TU München 1995
- [73] Pohl, H-Ch.: Logistikmanagement. Funktionen und Instrumente. Implementierung der Logistikkonzepte in und zwischen Unternehmen, Berlin, Springer-Verlag 1994
- [74] Baumgarten, H.: Trends und Strategien in der Logistik 2000. Analysen – Potentiale – Perspektiven, TU Berlin, Institut für Technologie und Management, Bereich Logistik 1996

- [75] Weber, R.: Zeitgemäße Materialwirtschaft mit Lagerhaltung, 5. Auflage, Renningen-Malmsheim, Expert-Verlag 2000
- [76] Arnold, D.: Materialflusslehre, Braunschweig: Vieweg, 1995
- [77] Bürgel, W.: Vorlesungsskript Materialfluss- und Produktionslogistik, München, Lehrstuhl fml, TU München 1995
- [78] Kämmerling, W.: Materialflussplanung in der Fertigung, Düsseldorf: VDI-Verlag 1975
- [79] Lehmann, H.: Integrierte Materialfluss- und Layoutplanung durch Kopplung von CAD- und Ablaufsimulationssystem, Berlin, Springer 1997
- [80] Aggteleky, B.: Fabrikplanung: Werkentwicklung und Betriebsrationalisierung, Bd. 2 Betriebsanalyse und Feasibility-Studie, München, Hanser 1990
- [81] Wassermann, O.: Erfolgsfaktor Durchlaufzeiten, Köln, Verl. TÜV Rheinland, 1989
- [82] Nieß, P., Mader, E.: Bestände runter – aber wie ? Materialfluss, 25 (1994) Mai
- [83] Kuhn, A., Hellingrath, H.: Supply Chain Management, Berlin Heidelberg New York, Springer Verlag, 2002
- [84] Bauer, H.H., Grether, M., Brüsewitz, K.: Der Einsatz des Internet zur Vertriebsunterstützung im Automobilhandel, Reihe Management Know-how Nr. M46, Institut für Marktorientierte Unternehmensführung, Universität Mannheim 2000
- [85] Reichwald, R., Moslein, K., Sachenbacher, H., Englberger, H., Oldenbrug, S.: Telekooperation. Verteilte Arbeits- und Organisationsformen, Berlin: Springer 1998
- [86] Reichwald, R./ Rupprecht, M.: Einsatz von Informations- und Kommunikationstechnologien im Rahmen zwischenbetrieblicher Kommunikation, Hermanns, A., Felgel, V. (Hrsg.): Handbuch des Electronic Marketing, München 1992
- [87] Baumberger, J.; Gmür, U.; Käser, H.: Ausbreitung und Übernahme von Neuerungen. Ein Beitrag zur Diffusionsforschung, 2 Bände, Bern/ Stuttgart 1973

-
- [88] Österle, H.: Management und Informatik. Arbeitsberichte der Abteilung für Wirtschaftsinformatik der Hochschule St. Gallen 1986
- [89] Kasper, H.: Innovation in Organisationen. Konzeptionelle Arbeit mit empirischen Befunden, Wien 1980
- [90] Berthel, J.: Verhindern Führungsdefizite Innovationen? Innovationsorientierung in der Unternehmensführung, Zeitschrift Führung und Organisation, 56, 1987
- [91] Böhnisch, W.: Personale Innovationswiderstände, Handwörterbuch des Personalwesens, Hrsg. v. E. Gaugler, Stuttgart 1975
- [92] Scheer, A.-W.: CIM-Strategie als Teil der Unternehmensstrategie, Berlin 1990
- [93] Doetsch, E., Wolf, H.: CAI, Stuttgart 1988
- [94] Froitzheim, U. J.: Ans Netz gekettet, Wirtschaftswoche, 47. Jg., Nr. 39, 24.09.1993
- [95] Kudlich, T.: Optimierung von Materialflusssystemen mit Hilfe der Ablaufsimulation, Dissertation Technische Universität München, Fakultät für Maschinenwesen, 2000
- [96] Baecke-Heger, F.: Produktion zum Fließen bringen, Markt und Mittelstand, Februar 2004, Markt und Mittelstand Verlag, 2004
- [97] Wannenwetsch, H., Nicolai, S.: E-Supply-Chain-Management, Gabler Verlag, Wiesbaden 2002
- [98] Nußdorfer, R.: Das EAI-Buch, Selbstverlag, CSA-Consulting, 2000
- [99] Meyer, M.: CRM-Systeme mit EAI – Konzeption, Implementierung und Evaluation, Vieweg Verlagsgesellschaft, Braunschweig/Wiesbaden, 2002
- [100] Buhl, L., Christ, J., Pape, U.: Marktstudie: Softwaresysteme für Enterprise Application Integration, ALB/HNI-Verlagsschriftenreihe, Paderborn, 2001
- [101] Keller, W.: Enterprise Application Integration – Erfahrungen aus der Praxis, dpunkt.verlag, Heidelberg, 2002

- [102] Denert, E.: Software-Engineering, Springer Verlag, 1991
- [103] Scheels, I.: Programmieren, Markt+Technik Verlag, 2001
- [104] Goller, H.: Das LAMP Buch: Webserver mit Linux, Apache, MySQL und PHP, SuSE-Press, Nürnberg 2004
- [105] Yank, K.: PHP und MySQL, Dpunkt Verlag, 2002
- [106] Höfling, J.: Die Spreu trennt sich vom Weizen, IT-Management 2-2004, It-Verlag, 2004
- [107] Krause, J.: Programmieren lernen in PHP5, Hanser Verlag, 2004
- [108] Chorafas, D.: Integrating ERP, CRM, Supply Chain Management and Smart Materials, Auerbach, 2001
- [109] Robert France Group: Total Cost of Ownership for Linux in the Enterprise, 2002, <http://www.rfgonline.com>
- [110] Marx, B.: Linux Manager Guide: was Entscheider über das Betriebssystem und Open Source wissen müssen, SuSE-Press, Nürnberg 2001
- [111] Schröder, J., Müller, M.: Webserver betreiben – http und Apache: Grundlagen, Konzepte, Lösungen, Dpunkt Verlag, 1999
- [112] Wölfer, T.: Webserver Survival Guide, Franzis Verlag, 2002
- [113] Vogel, Ch.: Apache 2 – Kompendium, Markt+Technik Verlag, 2004
- [114] Werner, H.: Supply Chain Management: Grundlagen, Strategien, Instrumente und Controlling, Gabler Verlag, Wiesbaden, 2002
- [115] Wildemann, H.: Wirtschaftlichkeitsreserven im Umlaufvermögen, TCW Transfer-Centrum, München 2000
- [116] J. Emigh, Framingham: Vendor-managed inventory, 1999, Computerworld

- [117] Lobeck, F.: Konzept zur Optimierung von Produktentwicklungsprozessen einschließlich Simulation und Rapid Prototyping unter Verwendung eines neuen PLM-CAD-Integrationsmoduls, Universität Duisburg-Essen, 2004
- [118] Scheer, A.-W., Abolhassan, F., Bosch, W.: Real-Time Enterprise, Springer-Verlag, 2003
- [119] Bullinger, H.-J., Berres, A.: E-Business-Handbuch für den Mittelstand: Grundlagen, Rezepte, Praxisberichte, Heidelberg et al., 2000
- [120] Fähnrich, K.-P.: eBusiness II: Vorlesung im Schwerpunkt Praktische Informatik / Angewandte Informatik, Wintersemester 2003/2004, Universität Leipzig, Anwendungsspezifische Informationssysteme, 2003
- [121] Wieland, Th.: XML mit PHP, Siemens AG, Corporate Technology, XMLOne München, 2001
- [122] Krause, J.: Im Werden begriffen: PHP 5 Beta 1: mehr Objektorientierung und XML iX – Magazin für professionelle Informationstechnik, September 2003, Heise Zeitschriften Verlag, Hannover 2003
- [123] Schneier, B.: Secrets & Lies. IT-Sicherheit in einer vernetzten Welt, Wiley dpunkt.verlag, 2001
- [124] Mühlenbrock, F.: IT-Sicherheit, Smartbooks, 2002
- [125] Busch, C., Wolthusen, S.: Netzwerksicherheit, Spektrum Akademischer Verlag, 2002
- [126] Samar, R., Stocker, C.: PHP de Luxe: Fortgeschrittene PHP-Programmierung, mitp-Verlag, Bonn 2002
- [127] Plattner, D.: Mittelstandsfinanzierung im Wandel: Finanzierungsstruktur und Ertragslage Mittelständischer Unternehmen 1998-2001, KfW-Research, Mittelstands- und Strukturpolitik No. 30, Oktober 2003
- [128] Streicher, H.: Enterprise Portals kanalisieren die Informationsflut, FIN.KOM, Magazin für Banking Innovation, 2/2000, Entory AG, Karlsbad-Ittersbach

-
- [129] Statistisches Bundesamt, Löhne und Gehälter, Arbeitskosten 1992 bis 2000, Deutschland
- [130] Bergers, D.: Management technischer Projekte und Geschäftsprozesse, Vorlesungsunterlagen, Universität-GH Essen, 2002
- [131] Brodie, M. L., Stonebraker, M. Migrating Legacy Systems, Morgan Kaufmann Publishers, 1995
- [132] Zahavi, R.: Enterprise Application Integration with CORBA: Component and Web-Based Solutions, Wiley Computer Publishing, 2000

11 Abkürzungsverzeichnis

4GL	Fourth Generation Language
A2A	Application to Application
ACL	Access Control List
ADE	Adapter Development Environment
AIC	Application Integration Components
AIX	Advanced Interactive Executive
AMI	Application Management Interface
ANX	Advanced Network Exchange
API	Application Programming Interface
APO	Advanced Planner and Optimizer
App	Application
APS	Advanced Planning System
ASP	Active Server Pages
ASCII	American Standard Code of Information Interchange
ATC	Application Transaction Coordinator
ATP	Available To Promise
B2B	Business to Business
B2C	Business to Customer
BDE	Betriebsdatenerfassung
CA	Computer Aided
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAI	Computer Aided Industry
CAM	Computer Aided Manufacturing
CAP	Computer Aided Production
CAS	Computer Aided Selling
CASE	Computer Aided Software Engineering
CAQ	Computer Aided Quality
CGI	Common Gateway Interface
CHAT	CrossWorlds Hybrid Asynchronous Transaction Model
CIC	Customer Interaction Center
CICS	Customer Information Control System
CIM	Computer Integrated Manufacturing
COBOL	COmmon Business Orientated Language
COM	Component Object Model

CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
CRM	Customer Relationship Management
CSV	Comma Separated Values
CTP	Capable To Promise
DB	Datenbank
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model
DFÜ	Datenfernübertragung
DOM	Document Object Model
DTC	Distributed Transaction Coordinator
DTD	Document Type Definition
EBCDIC	Extended Binary-Coded Decimal Interchange Code
EAI	Enterprise Application Integration
E-Business	Electronic Business
EC, e-commerce	electronic commerce
ECR	Efficient Consumer Response
EDI	Electronic Data Interchange
EDIFACT	Electronic Data Interchange for Administration
EDV	Elektronische Datenverarbeitung
EJB	Enterprise Java Beans
ENX	European Automotive Network Exchange
EP	Electronic Procurement
ERP	Enterprise Ressource Planning
ETL	Extraktion, Transformation und Laden
FTP, ftp	File Transfer Protocol
GUI	Graphical User Interface
GVO	Gruppenfreistellungsverordnung
HA	High Availability
HP-UX	Hewlett - Packard Unix
HTML	Hypertext Markup Language
HTTP, http	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol (SSL) Secured
IDoc	Intermediate Documents
IDL	Interface Definition Language
IMS	Information Management System
IIOP	Internet Inter-ORB Protocol
ISDN	Integrated Services Digital Network

ISO	International Organization for Standardisation
IP	Internet Protocol
IT	Informationstechnologie
JDBC	Java Database Connectivity
JDE	JD Edwards
JIS	Just In Sequence
JIT	Just In Time
JMS	Java Messaging Services
JSP	Java Server Pages
LAMP	Linux, Apache, MySQL, PHP
LAN	Local Area Network
LDAP	Light Weight Directory Access Protocol
MD5	Message Digest Algorithm 5
MIT	Massachusetts Institute of Technology
MOM	Message Orientated Middleware
MSMQ	MicroSoft Message Queue
NAFTA	North American Free Trade Agreement
NATO	North Atlantic Treaty Organisation
OASIS	Organization for Advancement of Structured Information Standards
ODBC	Open Database Connectivity
ODETTE	Organisation for Data Exchange by Tele Transmission in Europe
OEM	Original Equipment Manufacturer
OLAP	Online Analytical Processing
ORB	Object Request Broker
OS/390	Operating System /390 (IBM)
OS/400	Operating System /400 (IBM)
OTI	Open Transport Interface
P2P	Punkt-zu-Punkt
PCL	Programmable Command Language
PDM	Product Data Management
Perl	Page Extension and Reporting Language
PHP	Hypertext Preprocessor
PKI	Public Key Infrastructure
PPS	Produktionsplanung und –steuerung
PVCS	Polytron Version Control System
RMI	Remote Method Invocation
ROCE	Return on Capital Employed

RPC	Remote Procedure Call
RSA	Ron Rivest, Adi Shamir, Leonard Adelman
RTE	Real-Time Enterprise
SAP	Systemanalyse Programm-Entwicklung
SCM	Supply Chain Management
SCP	Supply Chain Planning
SCOR	Supply Chain Operations Reference-Modell
SDK	Software Development Kit
SI	Systemintegratoren
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SNA	Systems Network Architecture
SOA	Services Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SRC	Supply Reality Control
SSL	Secure Socket Layer
SUN	Stanford University Network
SuSE	Software und Systementwicklung
SWIFT	Society for Worldwide Interbank Financial Telecommunications
T&T	Tracking and Tracing
TCL	Tool Command Language
TCP	Transmission Control Protocol
TIB	The Information Bus
TP	Transaktionsprozess
TQM	Total Quality Management
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
VDA	Verband der deutschen Automobilindustrie
VMI	Vendor Managed Inventory
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WBEM	Web-Based Management
WindowsNT	Windows New Technology
WSDL	Web Services Description Language
X.25	Kommunikationsprotokoll für den Datenaustausch
X.31	Kommunikationsprotokoll für den Datenaustausch
XML	eXtensible Markup Language

XSDL	eXtensible Style Definition Language
XSL	eXtensible Stylesheet Language
XSLT	eXtensible Stylesheet Language Transformations
Y2k	Jahr-2000

12 Anhang

12.1 Tabellenbeschreibungen der Tabellen auth und rights der Datenbank konsi

Die Tabellenbeschreibung der Tabelle auth besteht aus vier Feldern, wobei das erste Feld id als maximal elfstelliger integer-Wert definiert ist (vgl. Abbildung 12.1). Die restlichen Felder username, firm_name und pword stellen character-Typen mit maximal 255 Zeichen Länge dar.

Feld	Typ	Attribute	Null	Standard	Extra
<u>id</u>	int(11)		Nein		auto_increment
uname	varchar(255)		Nein		
firm_name	varchar(255)		Nein		
pword	varchar(255)		Nein		

Abbildung 12.1: Tabellenbeschreibung der Datenbanktabelle auth

In der Tabellenbeschreibung der Tabelle rights sind drei Felder definiert, deren Werte aus elfstelligen integer-Werten (user) und 255-stelligen character-Zeichen bestehen können (Abbildung 12.2).

Feld	Typ	Attribute	Null	Standard	Extra
user	int(11)		Nein	0	
werk	varchar(255)		Nein		
lager	varchar(255)		Nein		

Abbildung 12.2: Tabellenbeschreibung der Datenbanktabelle rights

12.2 PHP-Programmskript zum VMI

Das erste Teilskript conf.php definiert globale Einstellungen (Name der Webseite). Der PHP-Parser erkennt den Beginn des PHP-Codes an der Zeichenkombination <? und das Ende an ?>, Kommentarzeilen werden mit einem doppelten Schrägstrich eingeleitet (vgl. Abbildung 12.3).

```
<?
// Definition der Webseite

$title = "TTB Lieferantenportal";

?>
```

Abbildung 12.3: Konfigurationsdatei conf.php

Im zweiten Teilskript conf.odbc.php) sind die Authentifizierungsdaten des Datenbankusers der iSeries-Datenbank (DB2) enthalten; bei erfolglosem Datenbankzugriff wird der Skriptaufruf abgebrochen (vgl. Abbildung 12.4).

```
<?
// Login-Informationen für DB2-Datenbank der iSeries

$dsn = "ASTTB";
$user = "QUSER";
$pass = "DATBAS2002";
$conn = odbc_connect ($dsn, $user, $pass) or die ( odbc_error ( ) );

?>
```

Abbildung 12.4: Konfigurationsdatei conf.odbc.php

Das dritte Teilskript conf.mysql.php enthält die Anmeldeinformationen des Datenbankusers der MySQL-Datenbank konsi, in der die Zugriffsrechte des Lieferanten auf die Lagerbestandsinformationen enthalten sind; bei erfolglosem Datenbankzugriff wird die weitere Skriptverarbeitung mit einer Fehlermeldung abgebrochen (vgl. Abbildung 12.5).

```
<?
// Login-Informationen für MySQL-Datenbank der LAMP-Umgebung

$this->conn = mysql_connect ( "localhost", "ttb", "ttbpassword" ) or die (mysql_error ( ) );
mysql_select_db ( "konsi" ) or die (mysql_error ( ) );

?>
```

Abbildung 12.5: Konfigurationsdatei conf.mysql.php

Die beiden Dateien head.php (vgl. Abbildung 12.6) und foot.php (vgl. Abbildung 12.7) dienen zur Definition des Tabellenlayouts, in dem die angezeigten Lagerbestandsinformationen dargestellt werden (Titel, Seitenfarben, Tabellen- und Tabellenzeilenaufbau), sowie des Tabellenanfangs und das –endes.

```
<?
// Seitenaufbau der Webseite mit Tabellendefinition

echo "\n<html>";
echo "\n<head>";
echo "\n<title>" . $title . "</title>";
echo "\n<style>";
echo "\nbody { background-color: #f1f3f8; font-family: Arial;
        scrollbar-arrow-color: #dadfec; scrollbar-base-color:#90a5c2; }" ;
echo "\nth    { background-color: #90a5c2; }";
echo "\ntr.zl1 { background-color: #dadfec; }";
echo "\ntr.zl2 { background-color: #bdc5dd; }";
echo "\na     { color: #800000; }";
echo "\n</style>";
echo "\n</head>";
echo "\n<body>";
echo "\n<table align=\"center\"><tr><td>";

?>
```

Abbildung 12.6: Konfigurationsdatei head.php

```
<?
// Abschluss des Tabellenlayouts

echo "\n</td></tr></table>";
echo "\n</body>";
echo "\n</html>";

?>
```

Abbildung 12.7: Konfigurationsdatei foot.php

Damit sind die Rahmendaten für das steuernde PHP-Skript index.php definiert, das mit der Zeichenkombination für den Parser-Aufruf <? Beginnt (Abbildung 12.8). Anschließend werden die zuvor definierten PHP-Dateien eingebunden und über Session-Variablen erfolgt

der weitere Programmablauf. Solange noch keine Session definiert ist, wird die default-Ausgabe generiert.

```
<?

// Ab hier beginnt die Einbindung von Session-Variablen; ansonsten wird das default-Modul am
// Seitenende aufgerufen

    session_start ( );

// Allgemeine Konfiguration: hier erfolgt die Einbindung der globalen Einstellungen aus der Datei
// conf.php

    require ("conf.php");

// MySQL-DB-Connect: Zugriff auf die Rechtestruktur des Lieferanten durch Einbindung der Datei
// conf.mysql.php

    require ("conf.mysql.php");
```

Abbildung 12.8: Beginn der Programmdatei index.php (Teil 1)

In der Abbildung 12.9 wird die Modulauswahl beschrieben.

```
// Modulauswahl: es stehen 4 Funktionsmodule zur Verfügung: Authentifizierung, Logout, Anzeige
// der Datenbankinhalte, Erzwingen der Authentifizierung

    switch ( $_get ["module"] ) {
```

Abbildung 12.9: Programmdatei index.php (Teil 2)

Das Modul login prüft, ob der Lieferant in der MySQL-Datenbank verschlüsselt eingetragen ist, übergibt bei erfolgreichem Login die notwendigen Session-Variablen und beendet die Skriptverarbeitung bei erfolglosem Login mit einer Fehlermeldung (vgl. Abbildung 12.10)

// Authentifizierung: Überprüfung, ob der Lieferant auf die Lagerinformationen zugreifen darf, mittels SQL

```
case "login":

    $query = "SELECT `id`, `uname`, `firm_name`, `admin`
              FROM `auth`
              WHERE `uname` = '" . addslashes ($_post ["uname"]) . "'
              AND `pword` = '" . md5 ($_post ["pword"]) . "' ";
    $result = mysql_query ($query);
    $rows = mysql_num_rows ($result);

    if ( $rows > 0 )
    {
        $_session ["login_id"] = chop (mysql_result ($result, $i, "id") );
        $_session ["login_uname"] = chop (mysql_result ($result, $i, "uname") );
        $_session ["login_firm_name"] = chop (mysql_result ($result, $i, "firm_name") );
        $_session ["admin"] = chop (mysql_result ($result, $i, "admin") );
        $module = "show"; header ( "location: " . $_server ["php_self"] . "?module=" . $module );
    }

    else

    {
        $_session ["login_uname"] = $_post ["uname"];
        $_session ["message"] = "Authentifizierung fehlgeschlagen";
        header ( "location: " . $_server ["php_self"] );
    }

    break;
```

Abbildung 12.10: Programmdatei index.php (Teil 3)

Im Modul logout werden die Session-Variablen gelöscht und neu gesetzt (vgl. Abbildung 12.11).

// Logout: Abmeldung von der MySQL- Datenbank

```
case "logout":

    unset ($_session ["login_id"]);
    unset ($_session ["login_uname"]);
    unset ($_session ["login_firm_name"]);
    unset ($_session ["admin"]);
    $_sesssion ["message"] = "Logout erfolgreich";
    header ( "location: " . $_server ["php_self"] );

    break;
```

Abbildung 12.11: Programmdatei index.php (Teil 4)

Die Anzeige der Datenbankinhalte ist nur nach erfolgter Anmeldung möglich, was im Teilskript in Abbildung 12.12 umgesetzt ist.

```
// Anzeige der Datenbankinhalte

case "show":

    // Zugang nur nach Authentifizierung

    if ( $_session ["login_id"] != "" ) {

        require ("conf.odbc.php");
        include ("head.php");
```

Abbildung 12.12: Programmdatei index.php (Teil 5)

Für den Fall, dass die Anmeldung erfolgreich war, erfolgt die Selektion der Lagerbestände, für die der Lieferant im Lieferantenstamm hinterlegt und für die er in der MySQL-Berechtigungsdatenbank freigeschaltet ist; zusätzlich wird über das Modul logout eine Logout-Möglichkeit dargestellt, über den die Abmeldung von der MySQL-Datenbank erfolgen kann (vgl. Abbildung 12.13).

```
// Auswahl der Berechtigung mittels SQL

$query = "SELECT `werk`, `lager`
        FROM `rights`
        WHERE `user` = " . addslashes ($_session ["login_id"]) . " ";
$result = mysql_query ($query);
$rows = mysql_num_rows ($result);

    if ( $rows > 0 ) {
        for ($i=0; $i < $rows; $i++){
            $werk = chop (mysql_result ($result, $i, "werk"));
            $lager = chop (mysql_result ($result, $i, "lager"));
            $daten [$i]["werk"] = $werk;
            $daten [$i]["lager"] = $lager;
        }
    }

// Definition des Seitenaufbaus und Selektion der Ausgabe

echo "\n<table border=\\"0\\" width=\\"70%\">";
echo "\n<tr>";
echo "\n<td align=\\"center\\" valign=\\"top\\" colspan=\\"10\">";
echo "\n<table border=\\"0\\" width=\\"100%\">";
echo "\n<tr>";
echo "\n<td>";
echo "\n<h2>" . $title . "</h2>";
echo "\n</td>";
echo "\n<td align=\\"right\\" valign=\\"top\">";
```

```

echo "\n<a href=\"\" . $_server ["php_self"] . "?module=logout\">
    <font size=\"2\">Logout</font></a>";
echo "\n</td>";
echo "\n</tr>";
echo "\n</table>";
echo "\n</td>";
echo "\n</tr>";

for ( $b=0; $b < count ($daten); $b++ ) {
    echo "\n<tr>";
    echo "\n<td colspan=\"4\">";
    echo "\n<font size=\"2\">";
    echo "\nWerk: <b>" . $daten [$b] ["werk"] . "</b>";
    echo "\n</font>";
    echo "\n</td>";
    echo "\n</tr>";

$query = "SELECT lgbs.lstnr, lgbs.lslgbe, teil.tebez1, teil.tebez2
FROM lgbs, teil
WHERE teil.tetenr = lgbs.lstnr
AND lgbs.lswknr = \" . $daten [$b] ["werk"] . \"
AND lgbs.lslanr = \" . $daten [$b] ["lager"] . \"
AND teil.tewknr = \" . $daten [$b] ["werk"] . \" ";

```

Abbildung 12.13: Programmdatei index.php (Teil 6)

Anschließend erfolgt die Datenbankabfrage auf der DB2-Datenbank, die bei leerem Inhalt eine Fehlermeldung liefert und ansonsten die Überschriften der tabellarischen Ergebnisreihen ausgibt (vgl. Abbildung 12.14).

```

// Ausführen des Query

$result = @odbc_exec ( $conn, $query );
$error = odbc_errormsg ();

if ( $error != "" ) {
    echo "\n<b>Fehler:</b><br><br>";
    echo $error;
}

else {

    // Auslesen der Ergebnisspalten

    $cols = odbc_num_fields ( $result );

    // Ausgabe der Kopfzeile

    echo "\n<tr>";
    echo "\n<th><font size=\"2\">";
    echo "\nTTB-Art.nr.";
    echo "\n</font></th>";
    echo "\n<th><font size=\"2\">";
    echo "\nBestand";
    echo "\n</font></th>";

```

```

echo "\n<th><font size=\"2\">";
echo "\nBezeichnung 1";
echo "\n</font></th><font size=\"2\">";
echo "\n<th><font size=\"2\">";
echo "\nBezeichnung 2";
echo "\n</font></th>";
echo "\n</tr>\n";

```

Abbildung 12.14: Programmdatei index.php (Teil 7)

Die Ausgabe der Ergebnisreihen sowie Definition des Tabellenlayouts mit Abschluss des Tabellenlayouts durch Einbindung der Datei foot.php ist in der Abbildung 12.15 dargestellt.

```

// Ausgabe der Ergebnisreihen
while ( odbc_fetch_row ( $result ) )
{
    // Ausgabe des Tabellenkörpers

    if ( $class == "z11" )
        $class = "z12";
    else
        $class = "z11";
    echo "\n<tr class=\"\" . $class . \"\">";
    echo "\n<td nowrap><font size=\"2\">";
    echo odbc_result ( $result, 1 );
    echo "\n&nbsp;</font></td>\n";
    echo "\n<td nowrap align=\"right\"><font size=\"2\">";
    echo number_format ( odbc_result ( $result, 2 ), 2, ',', ' ' );
    echo "\n&nbsp;</font></td>\n";
    echo "\n<td nowrap><font size=\"2\">";
    echo odbc_result ( $result, 3 );
    echo "\n&nbsp;</font></td>\n";
    echo "\n<td nowrap><font size=\"2\">";
    echo odbc_result ( $result, 4 );
    echo "\n&nbsp;</font></td>\n";
    echo "\n</tr>";
}

echo "\n<tr>\n";
echo "\n<td colspan=\"4\">\n";
echo "\n&nbsp;</td>\n";
echo "\n</tr>\n";
}
}
echo "\n</table>";
include ("foot.php");

```

Abbildung 12.15: Programmdatei index.php (Teil 8)

Wenn die Authentifizierung nicht erfolgreich war, erfolgt eine entsprechende Ausgabe (vgl. Abbildung 12.16).

```
// Erzwingen der Authentifizierung

    } else {
        $_session ["message"] = "Bitte Authentifizieren Sie sich!";
        header ( "Location: " . $_server ["php_self"] );
    }

break;
```

Abbildung 12.16: Programmdatei index.php (Teil 9)

Solange keine Session-Variablen existieren, wird als Default-Ergebnis die Login-Seite ausgegeben; dies verhindert den unberechtigten direkten Aufruf eines Moduls; das Skript wird beendet durch die Zeichenkombination `?>` (vgl. Abbildung 12.17).

```
// Default-Modul ist die Login-Seite

default:

include ("head.php");
echo "\n<form action=\"\" . $_server ["php_self"] . "?module=login\" method=\"post\">";
echo "\n<h2>" . $title . "</h2>";
echo "\n<font size=\"2\"><span style=\"width:140px\">Login:</span>";
echo "\n<input type=\"text\" size=\"30\" name=\"uname\" value=\"\" .
    $_session ["login_uname"] . "\">";
echo "\n<br>";
echo "\n<span style=\"width:140px\">Passwort:</span>";
echo "\n<input type=\"password\" size=\"30\" name=\"pword\">";
echo "\n<br><br>";
echo "\n<span style=\"width:140px\"> <input type=\"submit\" name=\"send\"
    value=\"Anmelden\"> </span>";
echo "\n<span style=\"color:red;\">";
echo $_session ["message"];
echo "\n</span>";
unset ( $_session ["message"] );
echo "\n</font></form>";
include ("foot.php");
unset ( $_session ["login_uname"] );
unset ( $_session ["login_id"] );

break;
}

?>
```

Abbildung 12.17: Programmdatei index.php (Teil 10)

12.3 PHP-Programmskript zum Auslesen der Oracle-Datenbank

Das erste Teilskript definiert eine Oracle-Umgebungsvariable und enthält die Anmeldedaten für die Oracle-Datenbank (vgl. Abbildung 12.18).

```
<?
// Setzen der Oracle-Umgebungsvariable
    PutEnv ("ORACLE_HOME=/opt/oracle");

// Login-Informationen für die Oracle-Datenbank

    $user = "QSYS";
    $pass = "QSYS";
    $dsn = "CAQ";

// MySQL-DB-Connect zur Oracle-Datenbank

    $conn = oci_plogon ($user, $pass, $dsn) or die (mysql_error ( ) ) ;
```

Abbildung 12.18: Programmdatei oci.php (Teil 1)

Anschließend werden der Name der Webseite definiert und mittels SQL die Selektion der Teilenummern aus der Oracle-Datenbank durchgeführt (vgl. Abbildung 12.19).

```
// Definition der Webseite

    echo "<h2>Teilenummern Feinplanungssystem</h2>";

// SQL-Query und Ausführung des Query

    $query = "SELECT * FROM artikel WHERE ROWNUM < 10";
    $result = oci_parse ($conn, $query);
    oci_execute ($result);

// Auslesen der Tabellenspalten

    $cols = oci_numcols ($result);
```

Abbildung 12.19: Programmdatei oci.php (Teil 2)

Es folgt die tabellarische Ausgabe der Ergebnisreihen bzgl. der Artikelinformationen aus der Oracle-Datenbank (vgl. Abbildung 12.20).


```
// Ausgabe des Tabellenkopfes

echo "\n<table border='1'\n">\n";

// Ausgabe der Kopfzeile

echo "\n<tr>";
for ( $i = 1; $i <= $cols; $i++ )
{
    $column_name = ocicolumnname ($result,$i);
    $column_type = ocicolumntype ($result,$i);
    $column_size = ocicolumnsize ($result,$i);
    echo "\n<td nowrap>";
    echo $column_name." / ".$column_type."(".$column_size.)";
    echo "\n</td>";
}

echo "\n</tr>";

// Ausgabe der Ergebnisreihen

while (ocifetchinto ($result, $row, oci_num + oci_return_nulls))
{
    echo "\n<tr style='background: $background;\n">";
    for ($i=0; $i < $cols; $i++)
    {
        echo "\n<td nowrap style='font-family:Courier;\n">";
        echo $row [$i];
        echo "\n&nbsp;</td>\n";
    }

    echo "\n</tr>";
}

// Abschluss des Tabellenlayouts

echo "\n</table>";

?>
```

Abbildung 12.20: Programmdatei oci.php (Teil 3)

12.4 PHP-Programmskript für die Einbindung von Bildinformationen

Falls Bilddateien zu einem Artikel vorhanden sind (Aufbau: artikelnummer.jpg), so wird das Bild von einem ftp-Server geladen und ausgegeben; alternativ dazu werden auch Bilddateien vom Typ png ausgegeben (Aufbau: artikelnummer.png). Ist kein Bildmaterial vorhanden (jpg oder png), so erfolgt eine entsprechende textlich aufbereitete Fehlermeldung (vgl. Abbildung 12.21).

```
<?
// Ort für die Bilddateien ist ein ftp-Server

if ( $_get ["artnr"] != "" )
{
    $handle = @fopen ("ftp://192.168.1.201/pics/" . $_get ["artnr"] . ".jpg", "r");

    // unterstützt werden jpg-Dateien ...

    if ( $handle )
    {
        header ("content-disposition: inline; filename=pic.jpg");
        header ("content-type: image/jpg");
        while (!feof ($handle))
        {
            echo fgets ($handle, 4096);
        }
    }
    else
    {
        // ... oder png-Dateien

        header ("content-disposition: inline; filename=pic.png");
        header ("content-type: image/png");
        $im = @imagecreate (250, 100) or die ("Kann keinen neuen GD-Bild-Stream erzeugen");
        $background_color = imagecolorallocate ($im, 230, 230, 230);
        $text_color = imagecolorallocate ($im, 0, 0, 0);
        imagestring ($im, 5, 10, 10, "Kein Bild vorhanden", $text_color);
        imagepng ($im);
    }
}

?>
```

Abbildung 12.21: Programmskript zur Einbindung von Bilddateien

12.5 PHP-Programmskript für die Aufbereitung der Lagerbestandsinformationen aus der ERP-Datenbank als XML-Nachricht

Die Lagerbestandsinformationen aus der DB2-Datenbank werden in eine XML-Nachricht konvertiert (vgl. Abbildung 12.22).

```
<?
// XML-Basisdaten

$xml = "<?xml version='1.0'?">\n";
$xml .= "<konsilager>\n";

// XML-Aufbereitung und Ausgabe der Ergebnisreihen

while ( odbc_fetch_row ( $result ) )
{
    $xml .= "<artikel nr='";
    $xml .= odbc_result ( $result, 1 );
    $xml .= "'>\n";
    $xml .= "<bestand>";
    $xml .= odbc_result ( $result, 2 );
    $xml .= "</bestand>\n";
    $xml .= "<bezeichnungen>\n";
    $xml .= "<bezeichnung nr='1'>\n";
    $xml .= utf8_decode (odbc_result ( $result, 3 ));
    $xml .= "</bezeichnung>\n";
    $xml .= "<bezeichnung nr='2'>\n";
    $xml .= utf8_decode (odbc_result ( $result, 4 ));
    $xml .= "</bezeichnung>\n";
    $xml .= "</bezeichnungen>\n";
    $xml .= "</artikel>";
}
$xml .= "</konsilager>\n";

?>
```

Abbildung 12.22: PHP-Skript zur Ausgabe der Datenbankinhalte als XML-Nachricht

12.6 PHP-Programmskript für den Versand der XML-Nachricht als Email

Die XML-Nachricht wird unter Einbindung der Dateien Mail.php und mime.php, die die mail-Funktion von PHP bereitstellen, als Email mit dem Betreff „XML-Nachricht“ MIME-konform formatiert und als Anhang an die Email-Adresse xmlmaster@ttb-group.de gesendet (Abbildung 12.23). Mail.php und mime.php sind PHP-Dateien aus den frei verfügbaren Mail- bzw. Mail_Mime-Packages der PEAR-Group (<http://pear.php.net>) und stellen grundlegende Mailinterfaces zur Verfügung, die den Versand von Emails mittels PHP unterstützen. Sie stehen unter <http://pear.php.net/package/Mail> bzw. http://pear.php.net/package/Mail_Mime im Internet zum Download bereit.

```
<?
( $_get ["mail"] )
{

    include('Mail.php');
    include('Mail/mime.php');

    // Definition des Mailaufbaus

    $text = 'XML-File im Anhang';
    $crlf = "\r\n";
    $hdrs = array ('Von' => 'xmlmaster@ttb-group.de', 'Subject' => 'XML-Nachricht');
    $mime = &new Mail_mime ($crlf);
    $mime->settxtbody ($text);
    $mime->sethtmlbody ($text);
    $mime->addattachment ($xml, 'text/plain', 'data.xml', false);
    $body = $mime->get ();
    $hdrs = $mime->headers ($hdrs);
    $mail =& mail::factory ('mail');
    $mail->send ($_get ["mail"], $hdrs, $body);
}
?>
```

Abbildung 12.23: PHP-Programmskript zum Versand der XML-Nachricht als Email-Anhang